MASTER EN SCIENCES INFORMATIQUES

Analyse de l'intégration de services de sécurité dans le standard EDIFACT

Mat, Xavier

Award date:
1995

Awarding institution:
Universite de Namur

Analyse de l'intégration de services de sécurité dans le standard EDIFACT

Mémoire présenté pour l'obtention
du grade de Licencié et Maître en
Informatique par

Xavier MAT
1994 - 1995

Facultés Universitaires Notre-Dame de la Paix
INSTITUT D'INFORMATIQUE
Rue de Bruxelles 61 - 5000 NAMUR
Tél.081/72.41.11 - Telex 59222 facnam-b - Telefax 081/23.03.91

Analyse de l'intégration de services de sécurité dans le standard EDIFACT

MAT Xavier

Résumé

Ce travail analyse l'intégration formelle de la sécurité dans le standard EDIFACT. Nous donnons d'abord une brève introduction à la sécurité de l'EDI (Echange de Données Informatisé) ainsi qu'au standard de représentation de données UN/EDIFACT, dans laquelle nous insistons sur le besoin de sécurité de l'EDI et du standard EDIFACT en particulier. Ensuite, nous tentons de mettre en évidence les services de sécurité pertinents pour les échanges de documents commerciaux, les primitives de sécurité nécessaires à ces services et les protocoles qui apportent une solution à chacun de ces services. Finalement, nous présentons l'intégration de ces protocoles dans la structure EDIFACT et la série de recommandations des Nations Unies où les services de sécurité sont soit intégrés dans le message lui-même, soit fournis par un message séparé.

Abstract

This work analyses the formal integration of security in EDIFACT. We give first a brief introduction to EDI (Electronic Data Interchange) security and to UN/EDIFACT standard, in which we lay stress on the need of security in EDI and particularly in EDIFACT standard. Next, we attempt to underscore the relevant security services as part of the exchange of business information, the security primitives required in the services and the implementation of services. Finally, we produce schemes for embedding these solutions in EDIFACT structures and present the United Nations recommandation set where the security services can either be integrated into the message itself or provided by a separate message.

Mémoire de licence et maîtrise en Informatique
Juin 1995
Promoteur : J. Ramaekers

Au terme de ce travail de fin d'études, je tiens à remercier tout particulièrement les personnes suivantes :

- Dr. Marijke De Soete de la société Europay International qui m'a aidé à établir un plan de travail, qui m'a fourni tous les documents nécessaires et est restée attentive à mon égard jusqu'en fin d'année ainsi que la société Philips qui m'a gracieusement envoyé toute la documentation requise;

- Monsieur Ramaekers, promoteur de ce mémoire, qui a guidé l'avancement de mon mémoire tout au long de l'année et qui m'a prodigué ses précieux conseils notamment pour la rédaction du travail;

- Joël Hubin et Wu Suchun qui ont fait preuve d'une grande disponibilité et qui ont mis à ma disposition l'ensemble de la documentation dont ils disposent.

- Madame D'udekem-Gevers qui dispose d'une impressionante documentation sur la sécurité de l'EDI et qui m'a guidé vers la spécialiste belge en matière de sécurité, madame De Soete.

Introduction

Quand on remplace les systèmes véhiculant du papier par l'EDI, Electronic Data Interchange, il est évident que la sécurité et les aspects légaux relatifs à la version électronique doivent être au moins aussi efficaces que ceux des versions manuscrites. Actuellement, les technologies en matière de sécurité sont mûres pour être incorporées dans l'EDI et les aspects légaux pour la sécurité de l'EDI sont déjà couverts dans les nouvelles propositions d'interchange agreements.

L'initiative des Nations Unies connue sous le nom d'UN/EDIFACT, United Nations EDI For Administration, Commerce and Transport, a permis de faire un grand pas en direction de la facilitation du commerce international. Par le développement et la promotion de la préparation et de la transmission électroniques des documents commerciaux, une très large gamme de fonctions commerciales a été rationalisée, la qualité et l'efficacité ont été améliorées et, finalement, les coûts ont été réduits.

Puisque le standard EDIFACT traite d'échanges EDI pour le commerce, il faut fournir des services qui protègent les partenaires commerciaux ainsi que les actifs de chacun d'eux. La décision d'utiliser ou non des services de sécurité est une décision commerciale qui fait suite à l'évaluation des pertes potentielles qui peuvent se produire lors d'une altération accidentelle ou malicieuse du message. Sans des mécanismes de sécurité appropriés pour prévenir ou détecter ces altérations du message, l'utilisateur d'EDIFACT s'exposera à des risques inacceptables. Il est clair que la sécurité est une exigence commerciale majeure et que, par conséquent, elle doit être au coeur d'EDIFACT.

Ce mémoire est divisé en trois grandes parties :

Une *première partie* introductive composée de deux chapitres; le **premier chapitre** présente une introduction à l'EDI et au besoin de sécurité pour celui-ci. Pourquoi et comment faut-il gérer la sécurité d'un échange de données informatisées? Le **deuxième chapitre** est consacré au standard EDIFACT. On situe le standard EDI dans le fonctionnement global d'un système EDI, on décrit ensuite ce standard EDIFACT et on termine en montrant que le standard EDIFACT est le lieu privilégié pour l'introduction de la sécurité dans l'EDI.

Une *seconde partie* qui est consacrée aux éléments de sécurité; le **troisième chapitre** présente les services de sécurité. Un tel service décrit un aspect du système de sécurité vu par l'utilisateur et est donc conçu pour satisfaire les demandes naturelles de l'utilisateur. Pour rendre ces services, il nous faut des moyens en conséquence. En cela, la cryptologie va nous apporter une aide sérieuse par le biais des diverses techniques développées pour chiffrer des documents électroniques. Les mécanismes de sécurité - le plus important étant la signature digitale - font l'objet du **quatrième chapitre**. Finalement, on développe les protocoles de sécurité qui sont en fait des

solutions construites à l'aide de combinaisons astucieuses de mécanismes de sécurité en vue de remplir les services de sécurité demandés. Le **cinquième chapitre** livre un protocole particulier pour chaque service décrit au troisième chapitre.

Une *troisième partie* qui décrit l 'intégration des services de sécurité dans EDIFACT; le **sixième chapitre** propose de multiples schémas d'intégration de la sécurité dans EDIFACT, chaque schéma solutionnant un service de sécurité pour un niveau donné de la syntaxe EDIFACT, sur base d'une analyse des contraintes imposées par les protocoles de sécurité et par la syntaxe EDIFACT. A l'heure actuelle, seule la sécurité au niveau du Message EDIFACT, qui laisse de côté le service de confidentialité, a déjà fait l'objet de recommandations de la part des Nations Unies. Le **septième chapitre** détaille la première recommandation qui est basée sur une solution qui intègre la sécurité dans le Message lui-même et le **huitième chapitre** prend en charge la seconde recommandation des Nations Unies qui est complèmentaire de la première et qui prône une solution où la sécurité est séparée du Message.

Pour satisfaire aux récentes recommandations des instances administratives de l'institut d'informatique, ce mémoire se présente en recto-verso. Nous espérons que cela ne nuiera pas trop à la qualité de lecture du document.

PARTIE I :

INTRODUCTION

Cette première partie introductive a pour but de présenter le cadre général de travail dans lequel vient s'inscrire « l'intégration de la sécurité dans le standard EDIFACT ». On navigue à la frontière de deux mondes très différents, celui de la sécurité informatique et celui de l'EDI. Cette partie remplira son objectif si on s'attache à répondre aux questions suivantes : Qu'est-ce que l'EDI? Qu'est-ce que la sécurité de l'EDI? Qu'est-ce qu'un standard EDI? Quelle est la structure du standard EDIFACT? Pourquoi sécuriser EDIFACT?

Chapitre 1

Introduction à la sécurité de l'EDI

1. Qu'est-ce que l'EDI?

1.1 Définition de l'EDI

L'*Electronic Data Interchange* ou EDI peut se définir comme étant l'échange de données électroniques

- réalisé entre applications (ou bases de données) informatiques conçues indépendamment et de même niveau et
- effectué sur base de spécifications standardisées (ou convenues bilatéralement) de la représentation et généralement aussi de la structuration et surtout de la sémantique des données échangées.

Cette définition est plus complexe que celles habituellement fournies dans la littérature. Cette complexité a pour but de cerner les spécificités de l'EDI par rapport à des techniques informatiques voisines. [GEV, 93]

Une classification des fonctionnalités de l'EDI peut être proposée : elle est basée sur la **nature du contenu** de l'échange électronique. Sur base de ce critère, OSITOP et Fox distinguent quatre catégories d'EDI qui peuvent être synthétisées comme suit :

- Trade Data Interchange (TDI), c'est-à-dire échange lié à une **transaction commerciale générale** (dans le domaine de l'administration, du commerce ou du transport) telle qu'une commande, une facture, etc.
- Electronic Funds Transfert (EFT) autrement dit les **transfert de fonds électroniques** (largement liés aux transactions bancaires);
- Technical Data Interchange, c'est-à-dire les **échanges de données techniques** (dessins, plans) (Computer Aided Design/Computer Aided Manufacturing : CAD/CAM).
- EDI interactif (ou I-EDI) (par exemple, pour la réservation des billets d'avion).

Le terme EDI est actuellement souvent utilisé pour désigner uniquement la première catégorie. En effet, c'est cette classe d'EDI qui a dès à présent fait l'objet de développement de la part des Nations Unies et de reconnaissance de la part de l'ISO. Dans le cadre de ce mémoire, l'acronyme EDI fera à chaque fois référence à la définition suivante (de TDI) : « capacité d'échanger des messages formatés de manière standardisée entre les applications des ordinateurs des partenaires commerciaux et/ou administrations avec le minimum d'intervention manuelle ». Cette capacité est d'importance stratégique et économique et l'on prévoit qu'elle deviendra une nécessité vu que le transfert électronique de données est en train d'être largement accepté comme une alternative commerciale aux services postaux.

Pour comprendre la raison de l'introduction de l'EDI, il faut se rappeler la manière dont les gens s'échangent des documents jusqu'à présent. Depuis des siècles, cet échange se fait par le service de poste. Cela exige une grande consommation de papier et, surtout, un délai d'attente qui peut durer des jours voire plus d'une semaine si l'échange se fait avec un pays étranger (voir figure 1.1).

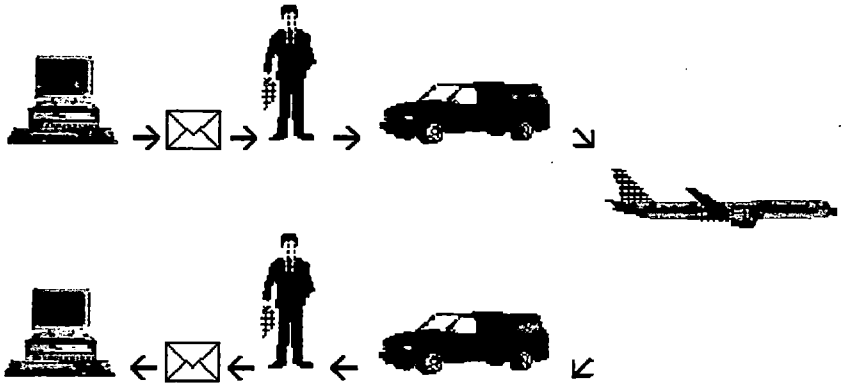

Figure 1.1 : Moyen classique de transfert de document

Alors que nous sommes à la fin du vingtième siècle, ce délai d'attente paraît bien déraisonnable. Puisque nous disposons aujourd'hui des moyens de télécommunications performants et relativement fiables, et puisque la plupart des documents commerciaux sont aujourd'hui générés par des applications informatiques, alors pourquoi ne pas faire communiquer des documents directement entre des applications informatiques par des moyens de télécommunication? (figure 1.2) En procédant de la sorte, on peut réduire le délai d'attente de jours en secondes, ce qui est un gain très considérable. C'est l'idée de départ qui a amené l'introduction de l'EDI.

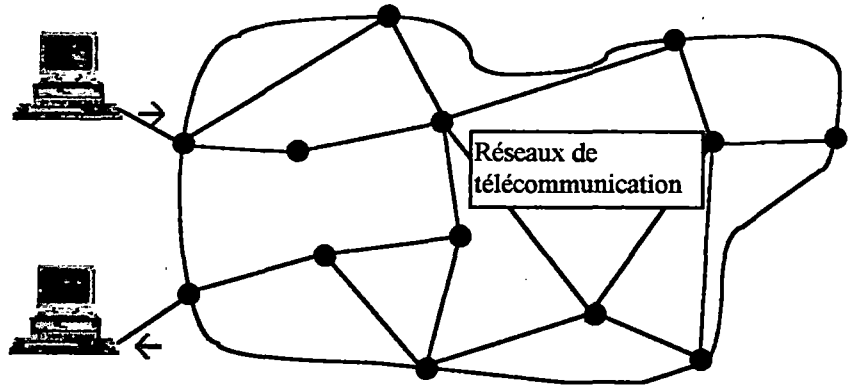

Figure 1.2 : Transfert de document par EDI

La commission des communautés européennes a reconnu l'importance de l'EDI et ceci se reflète dans l'effort qui se fait par l'intermédiaire de son programme sur les systèmes de commerce via les données électroniques (TEDIS).

Finalement, remarquons que la dernière catégorie (I-EDI) est seulement mise à l'étude pour le moment par les Nations Unies. L'intégration de la sécurité dans le futur EDI interactif sera abordée dans le chapitre 9 conclusions et perspectives.

1.2 L'étendue de l'EDI

Les messages qui sont actuellement échangés électroniquement incluent : bons de paiements des compagnies aux banques, bons de commandes pour marchandises et services, factures, relevés de compte, documents de souscription à une assurance, documents d'expédition et documents de dédouanement.

Pour effectuer un transfert, les organisations impliquées doivent se mettre d'accord sur les messages spécifiques et les formats qui vont supporter les données. Les problèmes de standardisation de la représentation des données feront l'objet du deuxième chapitre.

Pour effectuer un transfert de données, l'organisation ne sera pas la seule impliquée, on aura aussi :

* les fournisseurs de réseaux (ex: BELGACOM),
* les fournisseurs de services pour le réseau (souvent appelés Réseaux de Services à Valeur Ajoutée : RSVA - Value Added Networks Services: VANS) et
* dans certains environnements, des services de certification ;

Une autorité de certification est une tierce partie indépendante et en qui on a confiance, qui certifie à un utilisateur de réseaux qu'un autre utilisateur a été enregistré et possède tels ou tels attributs.

Toutes ses parties sont liées par une série de contrats, qui spécifient les services qui doivent être fournis, les coûts et les responsabilités de chacune des parties. Ces contrats sont vitaux pour un arrangement effectif. Cet aspect de l'EDI sera détaillé dans le point 4 (aspects légaux de l'EDI).

1.3 Bénéfices commerciaux dus à EDI

Les principaux bénéfices de l'utilisation effective de l'EDI sont :

- **Plus de rapidité dans la réalisation des transactions** : les fournisseurs ne doivent plus attendre des journées pour voir arriver une commande ou pour recevoir un paiement. Ceci permet au monde des affaires dans son ensemble de marcher sans à-coups. Dans beaucoup de secteurs commerciaux, le fait d'être capable de réagir très rapidement offre un avantage compétitif important.
- **Moins de paperasseries** : le coût des paperasseries impliquées dans les transactions internationales est estimé à 10% de la valeur de la marchandise. L'EDI peut aisément couper ce coût en deux.
- **Gains de temps** : l'EDI se charge de nombreuses tâches automatiques ou répétitives. En particulier, il réduit drastiquement le nombre de fois qu'une même information a besoin d'être recopiée ou retapée.

• **Moins d'erreurs** : les opérateurs qui se chargent d'entrer les données font des erreurs pour environ 2% des entrées. Par exemple, avec l'EDI, des nombres assez longs tels les codes de produits, qui ne signifie bien souvent rien à la personne qui le recopie sur un formulaire de commande, ne doivent pas être tapés au clavier pour chaque commande, ils sont stockés dans le système et utilisés en cas de besoin.

Bénéfices	Problèmes
Les transactions sont réalisées plus rapidement	Légaux
Moins de paperasseries	Sécurité
Gains de temps	Le besoin de standard
Moins d'erreurs	

Table 1.1 : Bénéfices et problèmes avec l'EDI.

La table 1.1 résume les bénéfices identifiés ci-dessus pour l'EDI. Pour récolter ces bénéfices, il faut cependant avoir conscience des problèmes qui se posent à l'EDI : les aspects légaux, la sécurité et le besoin de standard. Ces trois problèmes seront largement développés dans ce mémoire, en particulier, les problèmes de sécurité.

1.4 Les aspects légaux

Alors que les bénéfices apportés par l'EDI sont peu souvent disputés, il reste encore une large incertitude quant à son statut légal. Tous les systèmes légaux des pays de la communauté européenne sans exception ont été obligés de reconnaître la migration vers le commerce électronique. [CEC, 92]

Il y a une incertitude quant à la capacité de faire respecter les contrats établis par voie électronique. En cas de dispute, la recevabilité d'une preuve générée par ordinateur peut être difficile à établir. En fait, dans certains pays, les preuves générées par ordinateur pour certains types de transaction sont tout simplement refusées.

Le problème est exacerbé par l'absence de jurisprudence en rapport avec l'EDI.

Pour la plupart des organisations, les incertitudes légales ne devraient pas les empêcher de tirer avantage de l'EDI. Mais ils devront minimiser les risques légaux en se mettant d'accord avec les autres parties impliquées :

* sur les règles pour conclure un contrat;
* sur les responsabilités des différentes parties;
* sur les amendes pour avoir failli à ses responsabilités; et
* sur les procédures d'arbitrage qui devront être suivies en cas de désaccords.

Un document légal, connu sous le nom « interchange agreement », peut être utilisé par les partenaires commerciaux pour rencontrer les points ci-dessus. La chambre internationale du commerce a développé des règles uniformes de conduite lors d'échanges de données via les tétécommunications (UNCID-UNiform rules of Conduct for the Interchange of Data by teletransmission). L'UNCID, qui a été adopté par les Nations Unies, fournit une base pour établir des « interchange agreements ».

Les aspects légaux de l'EDI ne doivent être considérés isolément des aspects «sécurité»; les risques en matière de sécurité peuvent être atténués en adoptant des accords légaux qui identifient les responsabilités au point de vue sécurité et minimisent les risques légaux; inversément, les techniques de sécurité peuvent être utilisées pour établir une base légale à l'authenticité et pour réduire le risque de devoir recourir à de coûteux litiges.

1.5 Croissance de l'EDI

L'EDI est actuellement utilisé dans 50 secteurs industriels, incluant le secteur automobile, pharmaceutique, des épiceries et des soins de santé, et cette liste continue de croître. La figure 1.3 montre un résumé de la croissance soutenue de l'utilisation de l'EDI durant les six dernières années. [MAR, 93]

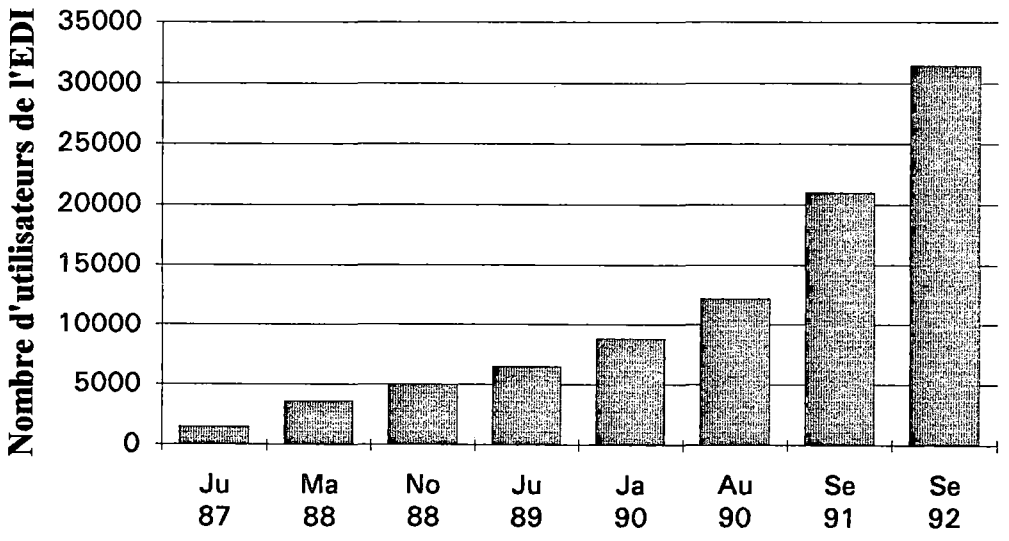

Figure 1.3 : Croissance mondiale de l'EDI

Le nombre d'utilisateurs dans le marché EDI a grossi de 2000 en juillet 1987 jusqu'à 31000 en 1992. Dans la période entre 1989 et 1992, la croissance annuelle du nombre d'utilisateurs EDI enregistrés n'est jamais descendue en dessous de 70%.

2 La sécurité de l'EDI

2.1 Introduction

Aujourd'hui, dans les systèmes qui travaillent avec du papier, on effectue plusieurs vérifications pour s'assurer que les fautes des copistes ont été détectées et corrigées. Par exemple, l'examen minutieux des pièces de papier par une équipe de copistes expérimentés identifie souvent des erreurs faites par le partenaire commercial. Dans un système EDI, il est nécessaire de remplacer ces procédures par de nouvelles procédures qui soient tout aussi performantes.

Le remplacement de ces procédures ne va pas seulement impliquer les partenaires commerciaux mais également les autres parties du système EDI. Quand les procédures de sécurité seront en place pour chacune des parties du système EDI, il sera nécessaire d'assurer que ces procédures, quand elles seront mises ensemble, sécuriseront l'ensemble. Chacune des parties contribuant au système entier a des priorités différentes - la sécurité est moins importante pour certaines organisations que pour d'autres. Lors de l'organisation de réseaux internationaux, il y a une myriade de détails techniques à résoudre simplement pour faire une connexion d'une organisation à une autre. La priorité sera donné à la réalisation des fonctionnalités du réseau et pas à sa sécurité même si la sécurité est un impératif commercial majeur pour toutes les parties.

Sur base de ce qui vient d'être dit, que va-t-on faire pour sécuriser un système EDI? on vient d'identifier un maillon assez faible du système EDI; le transit du message EDI dans un réseau. On se pose donc la question suivante : Quels sont les dangers qui menacent un message EDI en transit? Une recherche approfondie permet de dégager les dangers suivants : la présence d'émetteur frauduleux, l'intégrité compromise d'un message, la répudiation d'un message, la divulgation de données confidentielles, les messages retardés, les messages détournés du droit chemin, les pertes temporaires ou permanentes du service EDI.

Ensuite, on s'interroge sur la sécurité des mécanismes en amont et en aval de son transfert via le réseau, c'est-à-dire la création d'un message pour l'extérieur et les transformations des messages entrants : Va-t-on pouvoir sécuriser ces deux processus? Il faut, par exemple, assurer un contrôle d'accès pour le premier processus et une détection et correction des erreurs pour le deuxième processus.

Ces menaces et les possibles contre-mesures qui existent dans un environnement EDI seront discutées dans les chapitres suivants.

Cependant, la sécurité absolue est impossible, des erreurs peuvent se produire, les machines peuvent être en panne, les logiciels peuvent contenir des bugs. Donc, la gestion nécessite de déterminer comment on peut réaliser une sécurité acceptable en vue de faire du commerce tout en optimisant son coût. Le point 2 résume les différents risques de l'EDI pour une organisation et le point 3 décrit formellement la gestion de la sécurité.

2.2 Points à considérer quand on aborde la sécurité de l'EDI

Les questions clefs suivantes requièrent des réponses claires dans le but d'évaluer l'étendue des risques auxquels l'organisation s'expose dans le systèmes EDI proposé :

- Est-ce que l'organisation peut se compromettre dans des engagements commerciaux inacceptables en travaillant à partir des messages reçus ?
- Est-ce que tout le monde peut être assuré que les messages importants émis sont reçus par la personne à qui l'organisation les destinait et sont dans la forme exacte selon laquelle l'organisation voulait les envoyer?
- Qu'est-ce qui se passe si le système EDI est indisponible? Pour une heure? Un jour? Un mois?
- Est-ce que le contrat avec les partenaires commerciaux et les fournisseurs de service et d'équipement protège efficacement l'organisation? si non, quelles démarches ont été prises pour limiter l'exposition restante?
- Est-ce que le système que l'organisation implémente se conforme aux exigences de régularité (exigences fiscales, exigences en matière d'audit, ...)?
- Y-a-t-il des informations confidentielles? Est-ce que ces informations confidentielles sont gardées confidentielles à travers le réseau?

Lorsque les risques dégagés sont commercialement acceptables, l'organisation peut alors procéder avec confiance au ramassage des bénéfices de l'EDI.

2.3 Comment gérer la sécurité?

Dans le but d'obtenir une réponse de coût raisonnable aux différentes menaces techniques et humaines à la sécurité de l'EDI, les organisations devront publier une politique de sécurité qui fait apparaître clairement, à tous les membres et gestionnaires de l'organisation, l'attitude de l'organisation en matière de sécurité. On va pouvoir créer, découlant de cette politique, des procédures de sécurité plus détaillées permettant des implémentations spécifiques de l'EDI, de telle façon que les obligations endossées par l'organisation dans les contrats EDI pourront être remplies et qu'on pourra même le démontrer. La définition de la politique devra aller de pair avec une analyse de risques, en utilisant une méthodologie formelle d'analyse de risques. Dans le contexte de l'EDI, l'analyse de risques devra :

- établir la valeur des avoirs informationnels et des services,
- déterminer et évaluer les menaces réelles posées par les développements basés sur l'EDI,
- identifier les menaces qui sont sous le propre contrôle de l'organisation et celles qui peuvent uniquement être contrôlées par de tierces parties et
- introduire des mesures de coûts raisonnables pour minimiser les menaces.

L'analyse devra inclure l'identification des menaces qui découle de la confiance placée sur de tierces parties, tels les partenaires commerciaux et les fournisseurs de réseaux.

3 Conclusion

L'EDI est une opportunité, pour chaque organisation, de mettre la puissance des ordinateurs et des télécommunications à leur service. Dans le futur, la plupart des transactions commerciales, qui actuellement se servent du papier, seront exécutées en utilisant l'EDI. La capacité de construire des systèmes EDI, qui seront suffisamment sûrs pour gagner la confiance de chacun, est le challenge de toutes les personnes impliquées dans les systèmes actuels. Si la sécurité est assurée, alors les opportunités de bénéficier de l'utilisation de l'EDI sont énormes.

Chapitre 2

Le standard EDIFACT

1. Qu'est-ce qu'un standard EDI?

1.1 Structure du système EDI

Il y a normalement quatre composantes principales à un système EDI :

* un logiciel d'application;
* des standards pour les messages;
* un logiciel de traduction; et
* des méthodes de communication de données.

Quasiment tous les messages EDI ont pour origine et ont pour destination les applications informatiques du partenaire commercial. Les documents de commerce, par exemple, sont traités par des procédures pour bons de commandes ou par des systèmes de paiements d'accompte. L'utilisation de messages standards acceptés, règles qui déterminent comment les messages EDI sont construits et de telles manières qu'ils soient compréhensibles par le système informatique du récepteur, est le moyen par lequel les différences entre les applications informatiques utilisées par les différents partenaires commerciaux ne sont plus des barrières significatives. Le logiciel de traduction construit les messages EDI à partir des données du système d'application selon les standards de messages et traduit les messages reçus en formats compréhensibles par l'application informatique réceptrice.

Le système de communications de données peut s'étendre des liaisons par appel automatique sur bande magnétique sur les lignes téléphoniques publiques aux réseaux de données publiques fournies par les PTTs.

Dans beaucoup de cas, les liaisons entre parties se feront via les réseaux de services à valeurs ajoutées - RSVA. Dans certains pays, les RSVA sont une option attrayante parce qu'ils résolvent des problèmes causés par l'utilisation de systèmes de communications incompatibles et parce qu'ils simplifient la transmission de messages; à la place d'établir des liaisons pour communiquer avec tous les autres partenaires commerciaux, vous devez uniquement vous connecter à un RSVA. Comme des réseaux publics moins chers utilisant les standards internationaux deviennent disponibles, l'attraction des RSVA va diminuer.

Les standards intervenant dans l'EDI sont les standards **de représentations de données** et de **communication**. Le rapport entre ces deux standards ainsi que leurs fonctionnalités propres sont illustrés dans la figure ci-dessous. [GEV, 93]

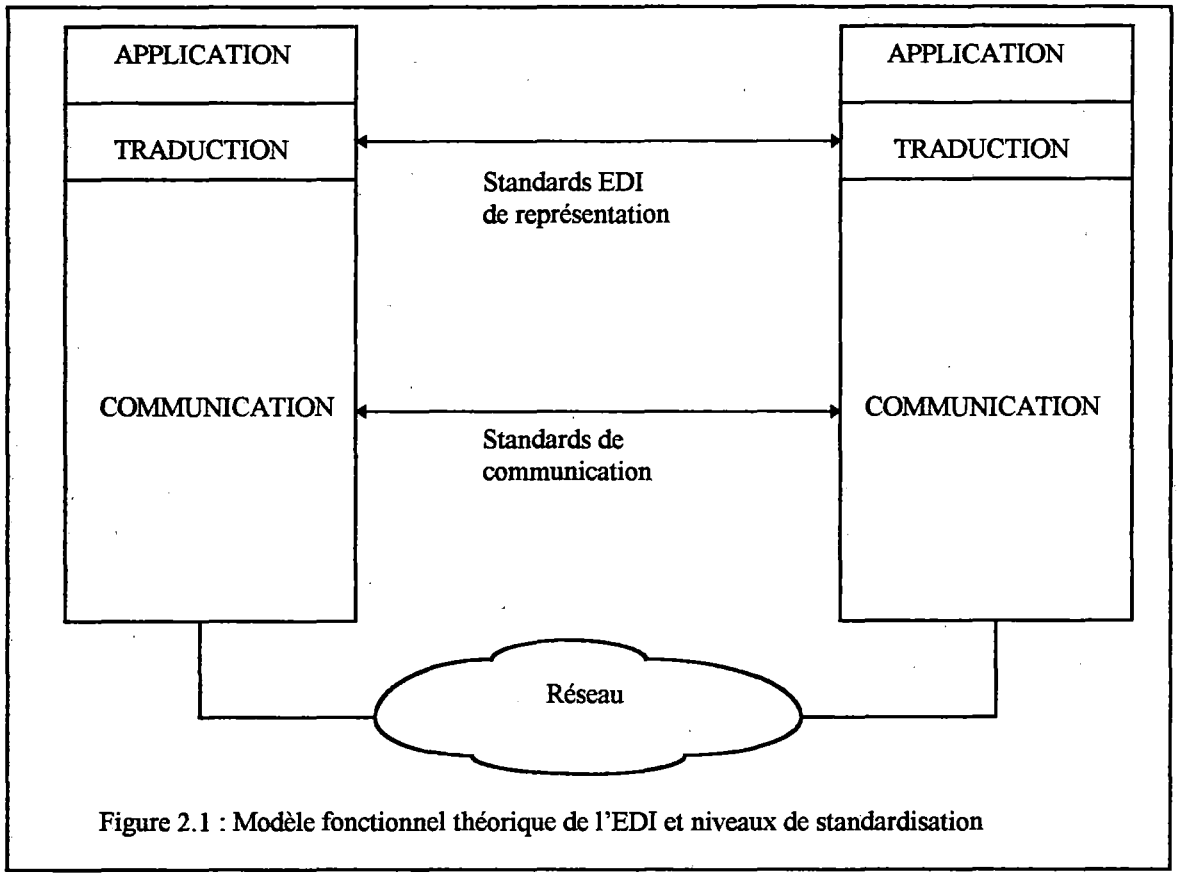

Figure 2.1 : Modèle fonctionnel théorique de l'EDI et niveaux de standardisation

La figure 2.1 présente un modèle basé sur le modèle de référence OSI (en couche). Nous supposerons le lecteur familier avec ce modèle; Dans l'autre hypothèse, vous pouvez toujours consulter [HEN,88] dans la littérature. Pour l'EDI, il convient de distinguer une hiérarchie de trois couches fonctionnelles par partenaire :

• La couche supérieure correspond à l'application de l'utilisateur final (gestion des commandes,...). Cette dernière n'est pas standardisée mais utilise un formalisme propre, qui ne peut être directement compris de son homologue. En revanche, elle doit utiliser des concepts (factures, commandes, etc.) partagés par la couche supérieure correspondante de son partenaire EDI.

• La couche moyenne, celle de traduction, dialogue avec son homologue grâce aux standards de représentations de données (ou plus simplement appelé standard EDI). Ces standards ne sont autres qu'un langage commun à plusieurs partenaires pour exprimer leurs messages.

• La couche inférieure, représentée ici de façon simplifiée, assure les fonctions nécessaires à la communication (accés au réseau, transfert, ...) et dialogue avec son homologue grâce à des standards de communication.

1.2 Les standards de représentation de données

Alors que les standards de communication concernent la façon dont les échanges sont réalisés, les standards de représentation sont relatifs au **contenu** des échanges EDI. C'est l'entreprise elle-même qui doit décider ce qu'il faut dire dans le message et qui doit exprimer ce message de façon telle que son partenaire EDI puisse en interpréter la signification. En bref, les standards de représentation concernent donc directement les

utilisateurs. Ils sont d'ailleurs faits par les utilisateurs et leur avenir individuel est totalement dépendant des utilisateurs (« user driven process »).

La fonction essentielle d'un standard est de réduire la variété. On comprend dès lors que des conventions de formats strictement limitées à deux partenaires commerciaux ne peuvent être considérées comme des standards. En revanche, il peut y avoir de l'EDI sans standard mais avec de simples conventions bilatérales fixant la façon d'exprimer les messages à échanger. Cette variété primitive d'EDI ne peut se concevoir que pour de l'EDI pratiqué avec peu de partenaires différents. Car si le nombre de partenaires (p) faisant ce type d'EDI augmente, très rapidement le nombre total de conventions à établir et à gérer (c) va s'accroître fortement. Dans l'hypothèse théorique où chaque partenaire fait de l'EDI avec tous les autres, on peut aisément calculer que ce nombre est : $c = p\,(p-1) / 2$. Ceci est illustré par la double figure 2. [GEV, 93]

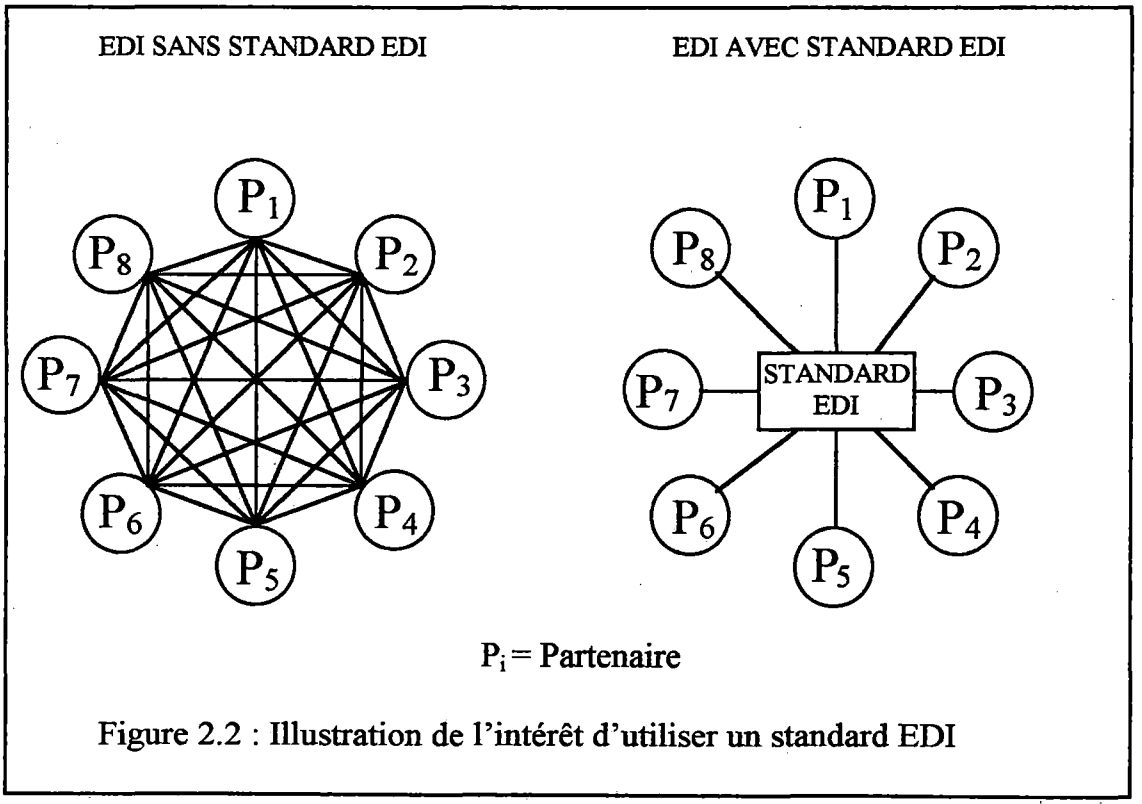

Figure 2.2 : Illustration de l'intérêt d'utiliser un standard EDI

Durant les années 70 et 80 furent développés des standards nationaux et spécifiques aux industries pour les messages. Les plus connus parmi eux sont :

- ODETTE (standard utilisé dans l'industrie de l'automobile)
- SWIFT (standard utilisé pour le transfert bancaire)
- UN-TDI (standard utilisé essentiellement en Europe)
- TRADACOMS (standard britannique)
- ANSI X.12 (standard américain)

Malheureusement, quand ces organisations commençèrent à étendre l'utilisation de l'EDI hors de leurs groupes industriels et hors de ses frontières nationales, ils découvrirent qu'ils ne pouvaient communiquer électroniquement parce qu'ils utilisaient une variété de standards incompatibles.

Ce problème d'incompatibilité fut la raison à l'établissement d'un ensemble de standards internationaux pour l'EDI, connu sous le nom UN/EDIFACT. Le standard

UN/EDIFACT est presque universellement reconnu comme étant le seul standard possible pour l'EDI international.

Inévitablement, dans des pays où l'EDI est déjà bien établi, les standards nationaux vont probablement continuer à être utilisés pour l'EDI domestique pour quelque temps encore. C'est notamment le cas des Etats-Unis où l'utilisation de ANSI X.12 est encore très répandue.

1.3 Les réseaux à valeur ajoutée (RVA) [MAR, 93]

Aujourd'hui, les compagnies utilisant des standards de communication et de représentation différents parviennent néanmoins à communiquer grâce aux services de RVA. Ce sont des vendeurs multiservices qui offrent une large gamme de services et de produits aux futurs utilisateurs d'EDI. Un RVA peut être utilisé de deux façons différentes : (1) comme une boîte aux lettres électroniques ou (2) pour les nombreux services qu'il est en mesure de fournir à un utilisateur EDI.

Les RVA permettent actuellement de répondre aux différents inconvénients de l'EDI, tels les problèmes de standards et de sécurité. Moyennant un coût élevé, ils reprennent à leur compte tous les aspects dont l'utilisateur final de l'EDI veut se décharger. Les bénéfices qu'une compagnie retire de l'utilisation des services d'un RVA incluent :

- Un lien de communications direct avec tous partenaires commerciaux.
- L'expérience et les connaissances du fournisseur de service du RVA des standards EDI existant et des technologies EDI impliquées.
- L'ampleur géographique et les économies d'échelle qu'un RVA fournit.
- La capacité d'un RVA de supporter de multiples formats de données.
- Les composantes à valeurs ajoutées - les services de formation et de consultance ainsi que les logiciels.
- Les services de boîte aux lettres des RVA, un système de messagerie store-and-forward.
- La capacité des RVA de supporter des protocoles et méthodes d'accès variées.
- Les RVA sont actifs toute la nuit et ils fournissent une transmission de messages 24 heures sur 24 en cas de besoin.
- La capacité des RVA de fournir des informations de contrôle qui permettent aux utilisateurs de vérifier les transmissions de messages et de documents avec leurs partenaires (audit).

Nous allons illustrer maintenant l'utilisation d'un RVA en temps que boîte aux lettres. Pour ce faire, nous allons comparer le fonctionnement d'un système EDI sans boîte aux lettres avec le fonctionnement d'un autre système EDI qui utiliseraient la boîte aux lettres. Le but d'un système EDI est de passer des données d'un ordinateur à un autre. Il fait ceci en un nombre d'étapes. Une première façon de faire est d'utiliser une connection point-à-point. Dans un réseau point-à-point, que l'on va opposer aux RVA, les partenaires commerciaux sont reliés directement. Si, par exemple, un fabricant de produits pharmaceutiques communique avec ses partenaires commerciaux via un réseau point-à-point, les étapes seront les suivantes :

1. Le système informatique du fabricant reçoit l'instruction de préparer trois messages EDI - un pour l'entrepôt, un pour les transports routiers et un pour le distributeur.
2. L'ordinateur du fabricant établit un lien de communication via un réseau téléphonique

public et envoie les messages EDI, qui sont étiquetés pour chaque partenaire commercial. Ce processus de transmission continue jusqu'à ce que tous les messages soient envoyés. Le système informatique du partenaire commercial doit être actif, capable de recevoir les signaux de communication en entrée (les appels) et, le plus important, tous les systèmes informatiques impliqués doivent être compatibles.

Ce système est approprié pour les organisations qui communiquent électroniquement seulement avec quelque partenaires commerciaux.

D'un autre côté, les RVA n'exigent pas des organisations qu'elles créent leurs propres sous-systèmes de communications et, de plus, les RVA peuvent fournir un réseau plus restreint et plus sûr. La figure 2.3 illustre un environnement de messagerie EDI qui incorpore un service de boîtes aux lettres électroniques d'un VAN-Value Added Network (RVA). Ce système de boîte aux lettres procure un avantage important, qui est que les deux entreprises peuvent travailler à leurs propres vitesses, sans avoir à se synchroniser. Ainsi, l'EDI réalise une fonction de tampon. [IMP, 93]

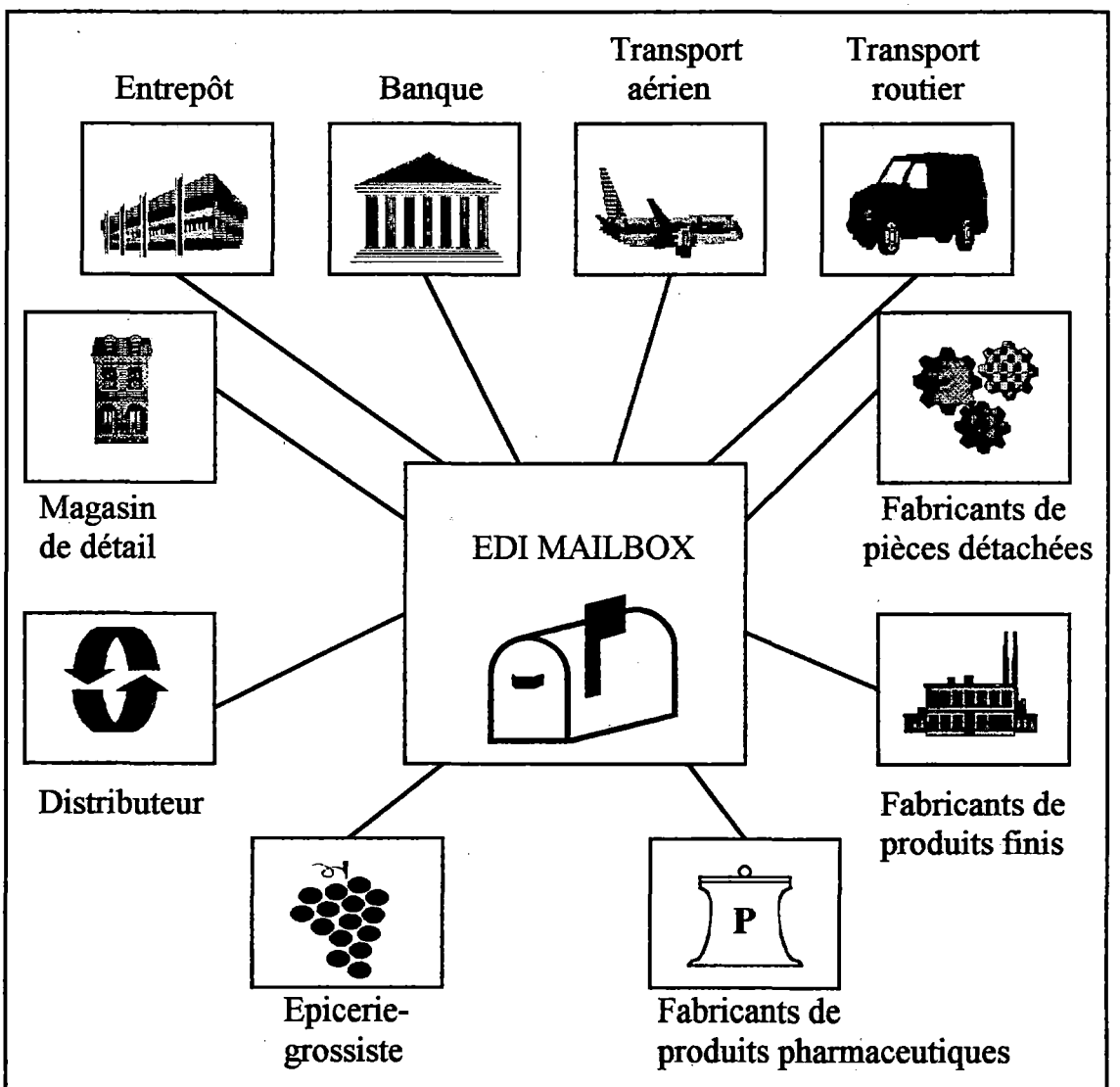

Figure 2.3 : EDI via un service de boîte aux lettres d'un VAN

Un fabricant de produits pharmaceutiques, qui veut échanger les trois même messages EDI, via ce système EDI particulier, doit se connecter à son « bureau EDI», un système

central d'ordinateurs possédé par un RVA, puis envoyer ses messages à son bureau EDI. Ils sont alors stockés dans la boîte aux lettres du destinataire, celle de l'entrepôt, du transporteur routier et du distributeur. Chacun de ces trois partenaires commerciaux peut aller chercher ses messages dans sa boîte aux lettres en se connectant à son bureau EDI. Ce service de boîte aux lettres fonctionne de manière comparable à celui d'une messagerie interpersonnelle de type X.400.

Les RVA les plus connus sont certainement les réseaux IBM, AT&T Easylink, BT Tymnet, INS, GEIS, EDS et McDonell Douglas. Ces réseaux EDI tendent vers une plus grande interconnectivité. Pour beaucoup de réseaux EDI, cela signifie qu'ils veulent établir des liens avec les réseaux X.400. Rappelons que le standard X.400 du CCITT est initialement conçu pour la transmission de messages de texte d'un utilisateur à un autre (messagerie inter-personnelle); cependant, une enveloppe spéciale, appelée PEDI ou X.435, a été conçue pour permettre à des messages EDI, qui ont une syntaxe différente, de passer sur le réseau X.400. Le standard X.435, qui est actuellement la méthode préférée pour transmettre des messages EDI sur les réseaux X.400, n'est pas encore largement utilisé. Mais il offre une opportunité aux systèmes EDI d'accéder à la grande population des utilisateurs X.400 et de faire usage des nombreux logiciels écrits pour ce standard.

2. Description du standard EDIFACT

2.1 Introduction

Il est nécessaire de présenter une description technique du standard EDIFACT pour comprendre où et comment l'on va intégrer la sécurité. Cette description ne sera pas exhaustive en ce qui concerne la syntaxe ISO-9735 mais elle poursuit simplement le but de permettre d'appréhender les termes techniques rencontrés dans les chapitres qui traitent de l'intégration de la sécurité dans le standard EDIFACT. [ISO-9735]

UN/EDIFACT ou plus simplement EDIFACT signifie United Nations / rules for Electronic Data Interchange For Administration, Commerce and Transport. Ce standard est né, en 1986, sous l'égide des Nations Unies et sur la base de longs et laborieux compromis entre deux standards préexistants : ANSI X.12 et UN-TDI. C'est en 1987 que le groupe de travail WP4 entérine l'acronyme EDIFACT et nomme trois rapporteurs, un pour l'europe de l'ouest (EDIFACT board), un pour l'amérique du nord (ASC X.12) et un pour l'europe de l'est (EDIFACT Comittee). Ceci implique une certaine lourdeur dans les processus de développement et de maintenance.

Le standard EDIFACT est composé d'une grammaire et d'un vocabulaire. Il est défini dans les documents suivants :

Grammaire

ISO 9735 - EDIFACT Syntax Rules :
Définition des règles de syntaxe de l'EDIFACT au niveau Application.

Vocabulaire

UN/EDIFACT Code Lists (EDCL) :
Définition des codes associés aux éléments de donnée simples.

UN/EDIFACT Data Elements Directory (EDED) :
Définition des éléments de donnée simples.

UN/EDIFACT Composite Data Elements Directory (EDCD) :
Définition des éléments de donnée composites.

UN/EDIFACT Standard Data Segments Directory (EDSD) :
Définition des segments de donnée standards utilisés pour les messages EDIFACT.

UN/EDIFACT Data Messages Directory (EDMD) :
Définition des messages standard EDIFACT spécifiés par l'UN/ECE (UNSMs).

UN/EDIFACT Syntax Implementation Guide :
Explications plus détaillées sur l'implémentation des règles de syntaxe.

UN/EDIFACT Message Design Guidelines :
Conseils à ceux qui voudraient créer de nouveaux messages EDIFACT.

2.2 Structure générale de l'interchange EDIFACT

Le fichier EDIFACT échangé entre deux partenaires est appelé l'« **interchange EDIFACT** ». L'interchange est un ensemble structuré et hiérarchique de messages EDI et de segments de service (« Unx ») comme le montre la figure 2.4.

Voici la signification de cette figure :

Une connexion contient un ou plusieurs interchange. Les protocoles techniques pour l'établissement et la terminaison ne font pas partie du standard international.

Un **interchange** EDIFACT est composé de **groupes fonctionnels** ou de **messages** indépendants sans groupe fonctionnel.

Un groupe fonctionnel comporte un ensemble de **messages**. Il sert à regrouper des messages du même type (ex : des factures) ou des messages de la même sous-adresse (l'adresse principale se trouvant dans l'en-tête de l'interchange).

Un message est composé de **segments** selon une structure hiérarchique. Les messages standards définis par l'UN/ECE (UNSMs) se trouvent dans le document EDMD.

Un segment est composé **d'éléments de donnée simples** et/ou **d'éléments de donnée composites**. Il y a deux types de segment : les segments de service (segments

« Unx ») et les segments de données. La définition des segments se trouve dans le document EDSD.

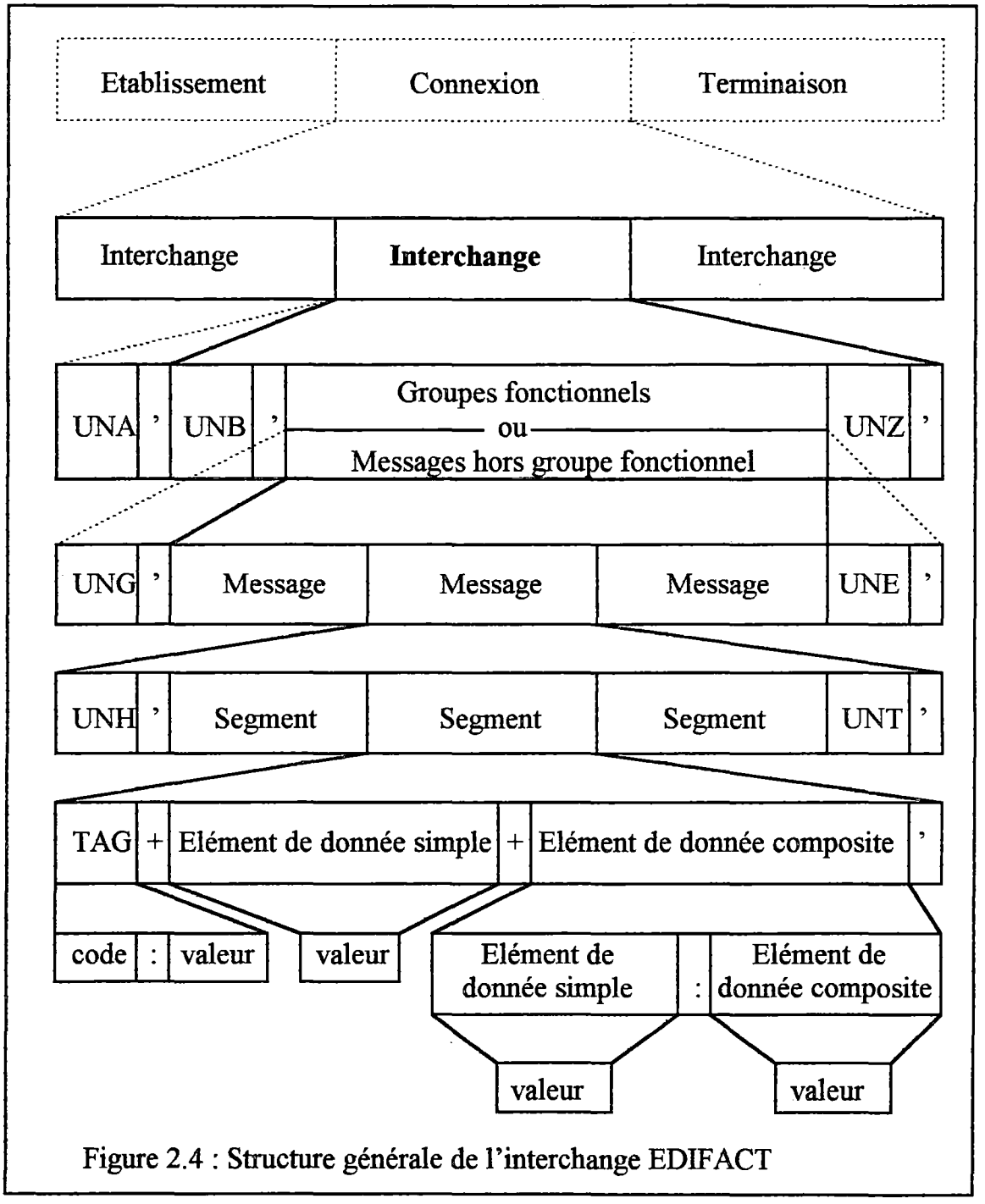

Figure 2.4 : Structure générale de l'interchange EDIFACT

Un élément de donnée composite est composé **d'éléments de donnée simples**. La définition des éléments de donnée composites se trouve dans le document EDCD.

Un élément de donnée simple est un champ de valeur. Il peut être de type alphabétique, numérique, alphanumérique ou « **code** identifiant ». La définition des éléments de donnée simples se trouve dans le document EDED.

Un code est l'identifiant d'une valeur choisie dans une liste de valeurs prédéfinies pour un élément de donnée simple particulier. Les listes des codes se trouvent dans le document EDCL.

Nous allons décrire ces composants de l'interchange EDIFACT en adoptant une approche « bottom-up », c'est-à-dire en partant du plus simple (code) pour arriver au plus complexe (message). Pour chacun de ces composants, nous donnerons un exemple concret, avec le codage associé défini par ISO 9735. Nous laisserons toutefois la description du **message** au paragraphe suivant (2.3) pour y discuter de l'important concept du diagramme de branchement.

2.3 Code

Pour les éléments de donnée simples de type « code identifiant », les valeurs possibles sont déterminés à l'avance par l'ISO. Ces valeurs sont alors codées et regroupées dans une liste des codes. Pour ces éléments de donnée, on est obligé de choisir un code dans la liste et on n'est pas autorisé à utiliser une valeur non-prédéfinie.

Prenons un exemple bien concret. L'élément de donnée simple «*Measure unit specifier*» est de type « code identifiant », il sert à spécifier l'unité de mesure utilisée. On trouve donc dans le document EDCL la liste des codes pour cet élément de donnée, liste dans laquelle l'ISO a répertorié toutes les unités de mesure imaginables et les a codées. En voici un petit extrait :

6411 Measure unit specifier

Unit name	code
Gram	GRM
Kilogram	KGM
Milligram	MGM
Day	DAY
Week	WEE
Year	ANN
Litre	LTR
Gallon	GLI
Cubic meter	MTQ

...

2.4 Elément de donnée simple

Un élément de donnée simple est un élément de donnée qui ne comporte qu'un seul champ de valeur. C'est par exemple le cas de l'élément de donnée « *Measure unit specifier* » déjà discuté plus haut. Voyons comment il est défini dans le document EDED :

6411 Measure unit specifier

Desc : In dication of the unit of measurement in which weight (mass), capacity, length, area, volume or oter quantity is expressed.

Repr : a3 Min : 3 Max : 3 Datatype : id

Tout comme dans la liste des codes, le nombre 6411 est la référence de l'élément de donnée et « Measure unit identifier» en est le nom. « Desc » est la description en anglais de l'élément de donnée. « Repr » indique la représentation, qui est ici alphanumérique de longueur fixe 3. « Min » et « Max » indiquent la longueur minimale et maximale. Et « datatype » indique le type, qui est ici « id » qui signifie « code identifiant ».

La raison de la double présence de l'élément de donnée « ***Measure unit specifier*** » dans EDED et dans EDCL est la suivante : EDED explique la signification de l'élément de donnée, avec la spécification de la représentation et du type. Si l'élément de donnée est du type « code identifiant », alors on peut retrouver dans EDCL la liste des codes qui le concerne.

Le codage d 'un élément de donnée simple est facile : on met sa valeur telle quelle, sans guillemets ou autres préfixes/suffixes pour les éléments de donnée du type alphabétique ou alphanumérique. Par exemple, si la valeur de « ***Measure unit specifier*** » est le kilogramme, alors son codage est KGM.

2.5 Elément de donnée composite

Un élément de donnée composite est un regroupement ordonné d'éléments de donnée simples formant un ensemble cohérent. Autrement dit, c'est un élément de donnée qui a plus d'un champ de valeur. [BLO, 91]

Prenons l'exemple suivant extrait d'EDCD :

C186 QUANTITY INFORMATION

Desc : Quantity information in a transaction, qualified when relevant.

Cont :

6063	Quantity qualifier	C	an..3	id	1	3
6060	Quantity	M	n..15	n	1	15
6411	Measure unit specifier	C	an..3	id	1	3

Il s'agit de l'élément de l'élément de donnée composite « QUANTITY INFORMATION » qui porte la référence C186. Il est composé de trois éléments de donnée simples présentés en colonnes : la première colonne indique la référence, la deuxième indique le nom, la troisième indique si le composant est obligatoire (M pour « Mandatory ») ou facultatif (C pour « Conditional »), la quatrième indique la représentation, la cinquième indique le type et les deux dernières indiquent les longueurs minimale et maximale.

Pour un élément de donnée composite, ISO 9735 spécifie la règle de codage suivante : on présente les valeurs des composants dans leur ordre d'apparition dans la définition, séparées par le caractère « : » (deux points). Par exemple, pour une quantité

facturée de 100 kilogrammes, on a « Quantity qualifier »=47, « Quantity »=100 et « Measure unit specifier »=KGM. Le codage de « QUANTITY INFORMATION » est donc:

47:100:KGM

2.6 Segment

Un segment est une suite ordonnée d'éléments de donnée simples ou composites. Le regroupement des éléments de donnée au sein des segments facilite la construction des messages EDIFACT, rend plus aisée la conception et la consultation des bases de données et facilite la saisie et l'édition des informations des documents. Les éléments de donnée regroupés dans un segment sont destinés à être traités en même temps et sont relatifs aux mêmes fonctions.

Prenons l'exmple suivant extrait de EDSD :

QVA QUANTITY VARIANCES

Function : To specify item details relating to variations between ordered/shipped and invoiced quantities.

C186 QUANTITY INFORMATION		M				
6063	Quantity Qualifier	C	an..3	id	1	3
6060	Quantity	M	n..15	n	1	15
6411	Measure unit specifier	C	an..3	id	1	3
4221 SHIPMENT/ORDER DISCREPANCY, CODED		C	an..2	id	1	2
6064 QUANTITY DIFFERENCE		C	n..15	n	1	15
C262 REASON FOR CHANGE		C				
4295	Change reason, coded	C	an..2	id	1	2
4294	Change reason	C	an..35	an	1	35

Pour un segment, ISO 9735 spécifie la règle de codage de base suivante : le TAG (étiquette) du segment doit apparaître en premier lieu. On présente ensuite les valeurs des différents éléments de donnée dans leur ordre d'apparition dans la définition, séparé par le caractère « + » (plus). Finalement, le segment doit se terminer par le caractère ' (apostrophe). ISO-9735 spécifie d'autres règles de codage de segment pour les cas où des composants facultatifs ou des éléments de donnée facultatifs sont absents. Nous n'allons pas décire ces règles dans le cadre de ce mémoire.

Nous allons prendre un cas concret. Imaginons qu'une société A commande à une société B 120 kilos d'un produit quelconque. La commande étant très importante, B ne peut liver que 100 kilos en ce moment. B envoie alors une facture EDIFACT de 100 kilos à A, avec le segment QVA mis aux valeurs suivantes :

« QUANTITY INFORMATION » :
« Quantity qualifier »=47 (=Quantité facturée)
« Quantity »=100
« Measure unit specifier »=KGM (=Kilogramme)

« SHIPMENT/ORDER DISCREPANCY, CODED »=BP (=Envoi partiel)
« QUANTITY DIFFERENCE »=20 (Différence = 20 kilos)
« REASON FOR CHANGE » :
« Change reason, coded »=QO (=Dû à la quantité commandée)
« Change reason »=Stock insuffisant (Commentaire)

Le codage de QVA est donc :

QVA+47:100:KGM+BP+20+QO:Stock insuffisant'

2.7 Segment de service

Il existe deux types de segment : les segments de données et les segments de service. Les segments de données sont ceux qui contiennent les informations internes au message (ex : noms et adresses, dates, articles, quantités, montants, valeurs, ...); le segment QVA en est un exemple. Les segments de service définissent le cadre et les caractéristiques de l'interchange EDIFACT. De plus, ils servent de délimiteurs pour séparer les interchanges, les groupes fonctionnels et les messages.

Les segments de services sont au nombre de huit. Ils portent tous un TAG qui commence par les lettres « UN » et ils sont définis dans EDSD de la mêm manière que pour les segments de données. Voici une brève description de ces segments de service :

Segment UNA : *Service string advice* (facultatif)

Le segment UNA permet de préciser les caractères de services utilisés. Ce sont les caractères de séparation, d'indication et d'échappement. La figure 2.5 montre les caractères de service par défaut de la syntaxe EDIFACT.

Fin de segment	' [apostrophe]
Séparateur entre éléments de donnée du segment et entre le TAG et le premier élément de donnée du segment	+ [plus]
Séparateur entre éléments constitutifs d'un élément de donnée composite	: [deux-points]
Caractère d'échappement	? [point d'interrogation]

Figure 2.5 : Caractères de services par défaut d'EDIFACT

Le caractère d'échappement, lorsqu'il précède un autre caractère, précise que ce dernier n'est pas un caractère de service mais bien un caractère de donnée.

Le segment UNA, s'il est présent au début de l'interchange, permet de modifier ces caractères de services utilisés par défaut.

Segment UNB : *Interchange header* (obligatoire)

Le segment UNB est le segment d'en-tête de l'interchange. Il sert à initialiser, identifier et préciser le cadre de l'interchange. Les principaux éléments de donnée du segment UNB sont :

- L'identification et le numéro de version de la syntaxe.
- L'identification de l'émetteur de l'interchange.
- L'identification du récepteur de l'interchange.
- La date et l'heure de préparation de l'interchange.
- La référence de contrôle de l'interchange.

Segment UNZ : *Interchange trailer* (obligatoire)

Le segment UNZ est le segment de fin de l'interchange. Il sert à terminer l'interchange et il peut être utilisé pour contrôler l'achèvement de l'interchange. Les éléments de donnée du segment UNZ sont :

- Le compteur du nombre de messages (ou de groupes fonctionnels) contenus dans l'interchange. Il permet de vérifier si tous les messages (ou groupes fonctionnels) sont bien reçus.
- La référence de contrôle de l'interchange. Cette référence est identique à celle du segment UNB.

Segment UNG : *Functional group header* (facultatif)

Le segment UNG est le segment d'en-tête du groupe fonctionnel, il sert à identifier le groupe fonctionnel.

Segment UNE : *Functional group trailer* (facultatif)

Le segment UNE est le segment de fin de groupe fonctionnel. Il sert à terminer le groupe fonctionnel et il peut être utilisé pour contrôler l'achévement du groupe fonctionnel.

Segment UNH : *Message header* (obligatoire)

Le segment UNH est le segment d'en-tête du message, il sert à identifier le message. Les principaux éléments de donnée du segment UNH sont :

- Le numéro de référence du message.
- L'identifaction du message. Précisions sur le type (ex : bon de commande), la version (ex : 90.2), etc.

Segment UNT : *Message trailer* (obligatoire)

Le segment UNT est le segment de fin du message. Il set à terminer le message et il peut être utilisé pour contrôler l'achèvement du message. Les éléments de donnée du segment UNT sont :

- Le compteur du nombre de segments contenus dans le message, y compris les segments UNH et UNT. Il permet de vérifier si tous les segments sont bien reçus.
- Le numéro de référence du message. Ce numéro est identique à celui du segment UNH.

Segment UNS : *Section contrôle* (facultatif, dépendant du type de message)

Le segment UNS est le segment de séparation entre les sections du message. En effet, certains types de message ont leurs segments de donnée séparés en trois sections : l'en-tête, le détail et le résumé. Le segment UNS comporte un seul élément de donnée qui est l'identification de la section.

	TAG	Intitulé du segment	Statut
	UNA	Avis de caractères de service	Facultatif
┌	UNB	En-tête d'interchange	Obligatoire
┌	UNG	En-tête de groupe fonctionnel	Facultatif
┌	UNH	En-tête de message	Obligatoire
		(Segments de donnée de la section d'en-tête)	
	UNS	Segment de séparation de section	Facultatif
		(Segments de donnée de la section de détail)	
	UNS	Segment de séparation de la section	Facultatif
		(Segments de donnée de la section de résumé)	
└	UNT	Fin de message	Obligatoire
└	UNE	Fin de groupe fonctionnel	Facultatif
└	UNZ	Fin d'interchange	Obligatoire

Figure 2.6 : Disposition des segments de service dans l'interchange.

Nous avons maintenant terminé la discussion sur les segments de service dans un interchange EDIFACT. La disposition de ces segments est résumée dans la figure 2.6.

2.8 Messages EDIFACT

Un message est une suite ordonnée de segments selon une structure hiérarchique à plusieurs niveaux d'imbrications. La définition des messages standards, c'est-à-dire la description des segments constituants des messages standards, se trouve dans le document EDMD.

Beaucoup de messages sont à l'étude à l'UN/ECE. Selon leur état d'avancement, ils portent un statut allant de 0 à 2. Voici la signification de ces statuts :

0 = Document provisoire (Draft document).

1 = Document provisoire pour essai formel (Draft for formal trial).
2 = recommandation (message standard des Nations unies ou UNSM).
Au cours de son développement, un type de messages doit passer par ces différents statuts et devenir finalement « stable » (statut 2). Lorsqu'un message atteint le statut 2, il est publié dans le document EDMD. Les messages de statut 2 les plus connus sont certainement la facture et le bon de commande. Chaque type de message porte un identifiant de six lettres, par exemple INVOIC pour la facture et ORDERS pour le bon de commande. Cet identifiant doit se trouver dans le segment UNH lors de l'envoi d'un message pour indiquer au récepteur le type de message.

En mars 93, 27 messages de statut 2 sont entérinés à l'UN/ECE et publiés. 25 messages de statut 1 et 111 messages de statut 0 sont à l'étude à l'UN/ECE. On peut consulter, dans [GEV, 93], la liste des types de messages EDIFACT en 1993.

Un message peut être défini de deux manières : par un diagramme de branchement ou par une table de segments.

2.8.1 Diagramme de branchement

Le diagramme de branchement est une représentation graphique qui définit un message EDIFACT en montrant les segments constituants et leur imbrication.

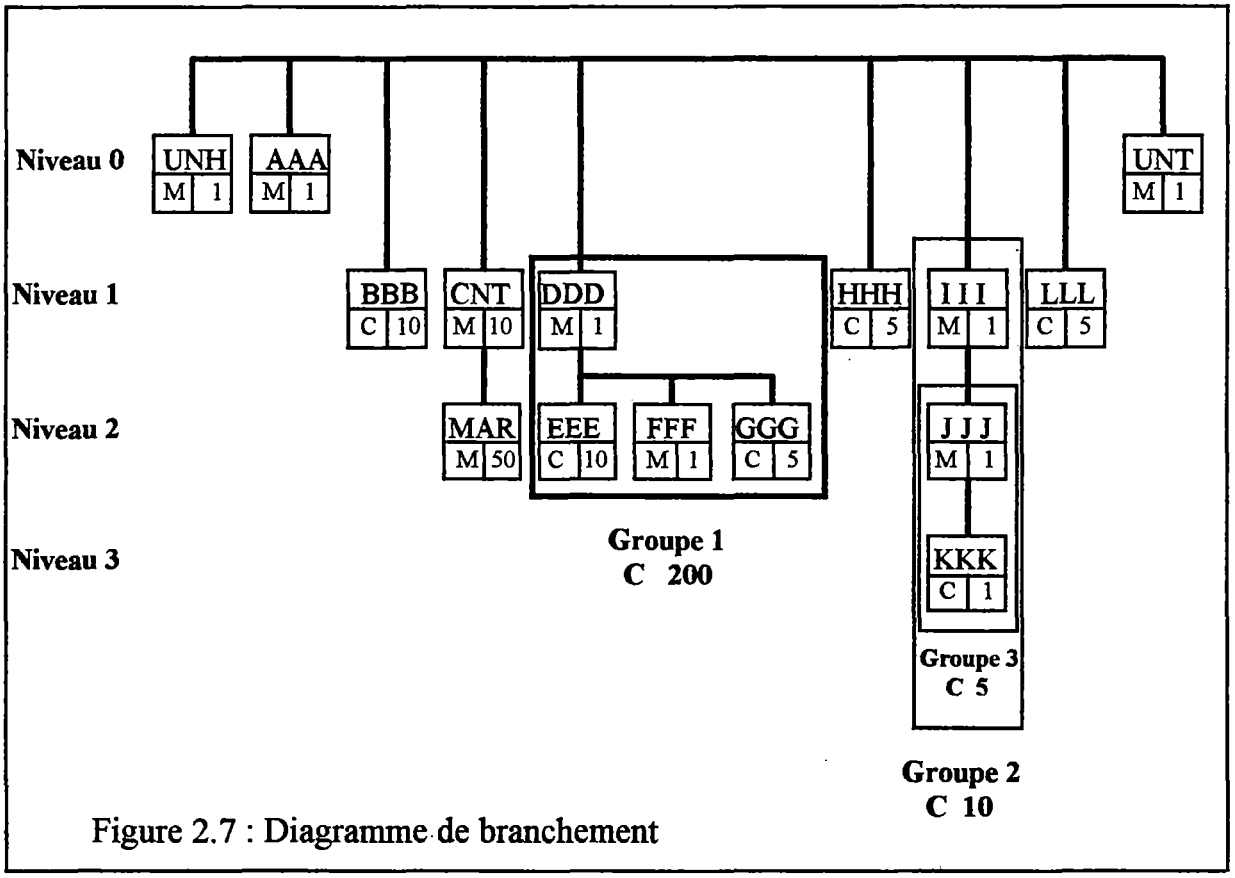

Figure 2.7 : Diagramme de branchement

La figure 2.7 montre un diagramme de branchement qui définit un message fictif. Il faut lire le diagramme de gauche à droite. Chaque rectangle du dessin représente un segment, avec des précisions sur le TAG, le statut (C=facultatif, M=obligatoire) et le

nombre de répétitions du segment. Par exemple, le segment AAA est un segment obligatoire non répétitif et le segment HHH est un segment facultatif qui peut se répéter au maximum 5 fois. Comme tout message, notre message fictif commence par un segment UNH et se termine par un segment UNT; tous les autres segments étant des segments de données qui contiendraient les informations du message.

Les segments sont présentés en plusieurs niveaux d'imbrication. Au niveau 0 se trouvent uniquement les segments non-répétitifs qui ne font pas partie d'une structure d'imbrication. Pour prendre un cas d'imbrication, regardons les segments CNT et MAR. Contrairement aux autres segments, ces deux segments ne sont pas fictifs. Un rapide coup d'oeil dans le document EDSD nous permet de connaître la signification de ces segments et d'imaginer la situation suivante : le segment CNT contiendrait les données décrivant le poids total et la valeur totale d'un conteneur d'une expédition de marchandises et le segment MAR contiendrait les données relatives au poids et à la valeur d'un seul type de marchandise pouvant être stocké dans le conteneur. On peut avoir au maximum 10 conteneurs et 50 types de marchandise par conteneur. Le diagramme implique que dans le codage du message, après l'apparition d'un segment CNT, il y aura 1 à 50 segments MAR qui suivent, le nombre de segments MAR étant le nombre de types de marchandise qui sont stockés dans le conteneur en question. Le segment CNT peut apparaître de 1 à 10 fois, le nombre d'apparitions étant égal au nombre de conteneurs expédiés. Nous voyons ainsi l'intérêt de l'imbrication : le segment MAR est imbriqué dans le segment CNT, car le premier est dépendant du dernier.

Les segments EEE, FFF et GGG sont imbriqués dans le segment DDD, leur père commun, appelé également **segment de contrôle**. Les quatre segments font partie d'un groupe, le groupe n°1, qui est facultatif avec un nombre de répétitions de 200. Cela implique que dans le codage du message, après l'apparition du segment DDD, il y aura 0 à 10 segments EEE, suivis nécessairement d'1 segment FFF, suivi de 0 à 5 segments GGG. Ce groupe n°1 peut lui-même apparaître de 0 à 200 fois consécutivement dans le message codé.

Les seul cas où des structures de groupe sont indispensables sont :
1) Lorsque le groupe est facultatif et que le segment de contrôle est obligatoire.
2) Lorsque le groupe est obligatoire et que le segment de contrôle est facultatif.
Le groupe n°1 de la figure 2.7 fait bien partie de ces cas. Sans la structure de groupe, il est tout simplement impossible d'enlever l'ambiguïté entre le caractère facultatif de l'ensemble des segments DDD-EEE-FFF-GGG et le caractère obligatoire du segment de contrôle DDD lorsqu'un des segments EEE, FFF et GGG est présent. Les groupes peuvent aussi être imbriqués comme le montrent les segments I I I, J J J et KKK de la figure 2.7. Pour le codage de ce cas-ci, après chaque apparition du segment I I I peuvent suivre les segments du groupe n°3 de 0 à 5 fois.

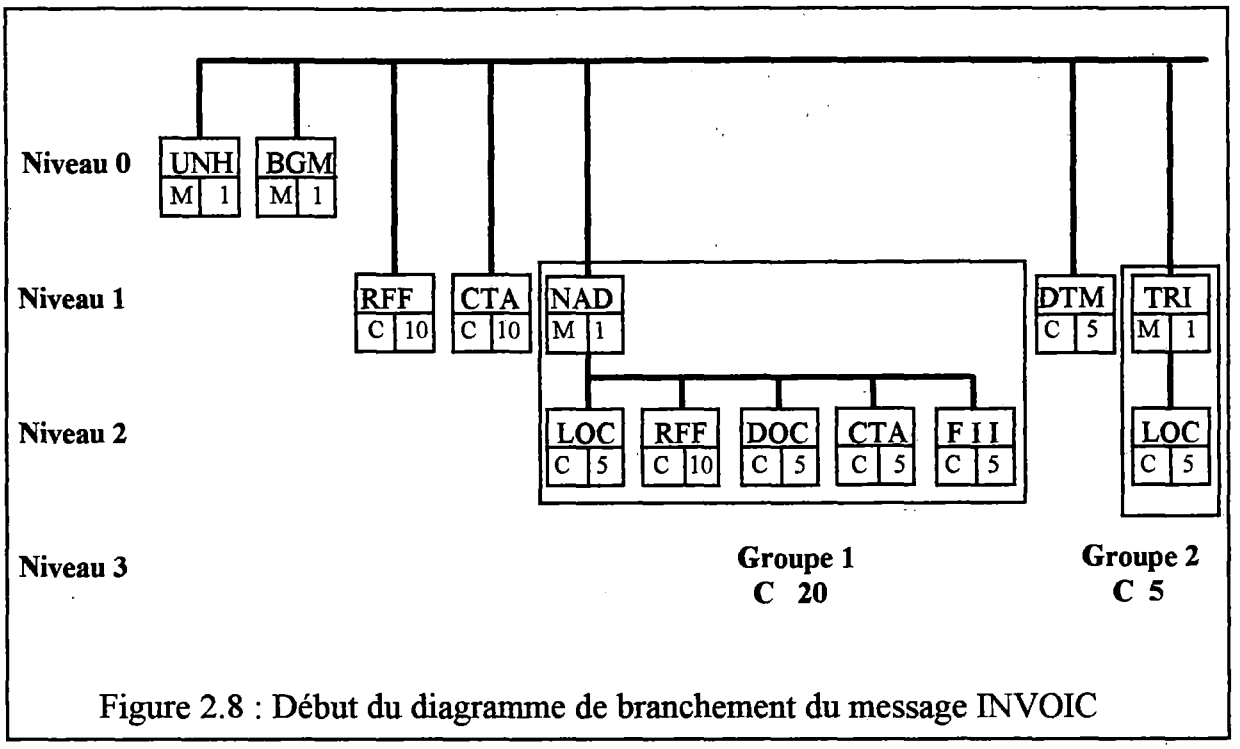

Figure 2.8 : Début du diagramme de branchement du message INVOIC

Les messages EDIFACT peuvent donc être définis à l'aide des diagrammes de branchement. En réalité, des messages tels que la facture ou le bon de commande ont des diagrammes beaucoup plus complexes que celui de notre message fictif. Nous montrons dans la figure 2.8 le début du diagramme de branchement du message INVOIC (facture); les lecteurs intéressés peuvent trouver dans [DID, 90] le diagramme complet.

2.8.2 Table de segments

La table de segments est une autre manière de définir un message. Il n'y a pas de différence fondamentale avec la définition par diagramme de branchement, si ce n'est que la présentation est ici textuelle et non graphique.

Nous donnons ici le début de la table de segments du message INVOIC, qui est l'équivalent du diagramme de branchement de la figure 2.8 :

UNH	Message header	M	1
BGM	Beginning of message	M	1
RFF	References	C	10
CTA	Contact	C	10
– Segment group 1		C	20
NAD	Name and address	M	1
LOC	Location identification	C	5
RFF	References	C	10
DOC	Documents required	C	5
CTA	Contact	C	5
F I I	Financial institution info	C	5
DTM	Date/time reference	C	5

– Segment group 2		C	5	⎤
TRI	Tax related information	M	1	⎥
LOC	Location identification	C	5	⎦

...

Chaque ligne correspond donc à un rectangle du diagramme de branchement, c'est-à-dire à un segment avec des précisions sur la TAG, le statut (C ou M) et le nombre de répétitions. Le premier segment qui apparaît dans un groupe est le segment de contrôle. Il est à noter que les niveaux d'imbrication ne sont pas mis en évidence dans la table de segments, mais on voit intuitivement que les segments qui suivent immédiatement le segment de contrôle sont à un niveau inférieur. Mais comment montrer une imbrication verticale de segments sans groupe comme les segments CNT et MAR de la figure 2.7? L'UN/ECE a laissé cette question sans réponse. Ce cas précis ne s'est jamais présenté dans aucune définition de message standard dont nous disposons, car dans un message réel, la quasi-totalité des structures d'imbrication à segment de contrôle obligatoire sont facultatives, donc nécessitant un groupement.

Si la table de segments a le mérite d'être plus facile à rédiger, le diagramme de branchement montre en revanche de manière plus nette les imbrications des segments et la structure générale du message.

2.8.3 Règles de codage du message

Le codage d'un message est une suite de segments codés selon les régles spécifiées dans 2.6. L'ordre d'apparition des segments dans le codage doit suivre les règles spécifiées dans 2.8.1, c'est-à-dire de gauche à droite (diagramme de branchement) ou de haut en bas (table des segments), et en profondeur d'abord. Les segments ne sont séparés par aucun caractère de séparation, si ce n'est le caractère de fin de segment ' (apostrophe). Lorsqu'un segment facultatif est absent, il est simplement omis dans le codage et même son TAG ne doit pas apparaître.

Les règles de codage d'un message sont donc assez simples. La figure 2.9 présente un exemple de codage d'un interchange EDIFACT contenant un message ORDERS. [GEV, 93]

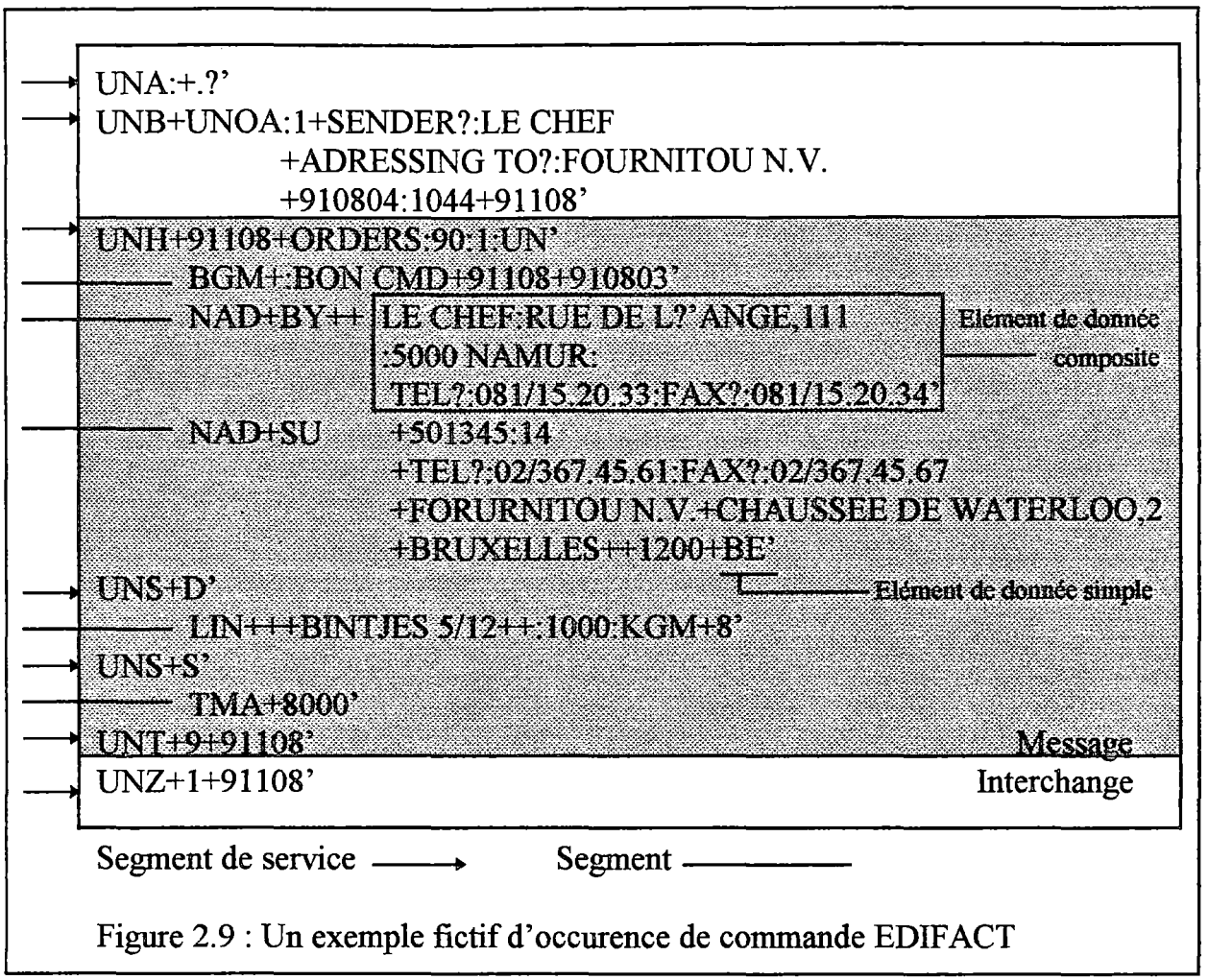

Figure 2.9 : Un exemple fictif d'occurence de commande EDIFACT

3. Pourquoi sécuriser EDIFACT?

3.1 Introduction

On a déjà établi la nécessité de sécuriser l'EDI dans le chapitre 1. Pour obtenir une motivation suffisante à l'intégration de la sécurité dans EDIFACT, il faut maintenant placer la dernière pièce du puzzle, le lien entre la sécurité et le standard EDIFACT. Par conséquent, pour répondre à la question « Pourquoi sécuriser EDIFACT? », il semble assez logique de commencer par montrer que, dans le système EDI complet, c'est au niveau du standard de représentation des données qu'il faut intégrer la sécurité (3.2) pour ensuite établir la non-sécurité du standard EDIFACT dans sa forme actuelle (3.3).

3.2 A quel niveau, dans le système EDI, faut-il réaliser la sécurité?

Les couches les plus importantes si l'on considère les aspects sécurités sont les couches supérieures, application et présentation, et les couches trois et quatre, les couches transport et réseau. Un certain nombre de services de sécurité peuvent être fournis par plusieurs de ses différentes couches, en particulier la confidentialité, on peut chiffrer entre chaque noeud du réseau ou de bout-en-bout. De façon similaire,

l'intégrité des données, ou l'authentification de messages, peuvent être, par exemple, offert par une spécification d'interface réseau tel X.25, plutôt que par le niveau application. Cependant, si ce service est offert par le réseau, il garantit uniquement que l'on détectera toutes les modifications des données durant la transmission. Il est impossible logiquement, en employant le réseau, d'être sûr que les données n'ont pas été modifiées par la suite.

Malgré que de nombreux standards et protocoles de communication soient sécurisés, il est nécessaire de sécuriser les messages, de bout-en-bout, indépendamment des nombreux moyens de transmissions. La raison en est qu'ainsi, la sécurité des messages peut être facilement maintenue et vérifiée, non seulement pendant le transfert dans les réseaux hétérogènes, mais également quand les messages sont dans les « log files » et dans les archives. Les concepts peuvent être appliqués aux messages de contrôle et d'accusé de réception pour étendre la gamme sécuritaire sur toute la transaction commerciale (incluant la non-répudiation de la réception, par exemple). Par conséquent, seules les couches présentations et applications seront pertinentes pour notre problème de sécurisation de l'EDI.

En conclusion, l'idée fondamentale est de sécuriser les messages EDI indépendamment de la façon de les transporter et de les stocker. Ceci présente des avantages manifestes, spécialement du point de vue légal et de vérification (audit). Puisqu'EDIFACT est un standard de la couche présentation /application, il est clair qu'en sécurisant les messages EDIFACT, on remplit les exigences de sécurité exposées ci-dessus.

3.3 La sécurité des standards EDI.

Quand les propositions des Nations-Unis pour un standard EDIFACT furent conçues, on ne donna que peu d'importance aux aspects de sécurité. En fait, cela se résumait à un segment d'authentification de 35 caractères. Ceci était clairement inspiré par les séries ANSI X12 concernant l'EDI en général, où un nombre de segments spécifiques aux services de sécurité étaient définis. Au contraire du standard EDIFACT, dans le standard ANSI X.12, des étapes pour introduire la sécurité ont été prises par l'intermédiaire du standard X12.58 (EDI Security Structure). Les buts en matière de sécurité de X12.58 furent de fournir l'intégrité, la confidentialité et la vérification de l'émetteur par le récepteur. En gros, l'idée est de rajouter des segments pour chacun des services de sécurité à appliquer. Il est évident, que la sécurité dans le monde ANSI était façonnée par la cryptographie conventionnelle, basée sur le chiffrement symétrique. Le revers le plus important lorque l'on applique ces systèmes est que les facilités de sécurité telles la preuve de l'origine et la non-répudiation ne sont pas possible pour des raisons logiques sans l'interaction d'une tierce partie.

En conclusion des points 3.2 et 3.3, il est propice de faire état du fait que, à moins d'utiliser PEDI, la chaîne manquante dans le monde EDIFACT est la contrepartie à X12.58. Si on utilise PEDI, une sécurité totale peut être mise en place au niveau Interchange (mais pas au niveau Message). Le but de ce mémoire est de combler le fossé si PEDI n'est pas utilisé ou si la sécurité est demandée au niveau message (plutôt qu'au niveau Interchange).

4. Conclusion

Pour clôturer cette première partie introductive, je voudrais faire deux observations d'ordre politique.

Premièrement, le chapitre 1 nous dit que la sécurité absolue est un mythe. Et pourtant, dans le même temps, les transferts financiers par voie électronique se généralisent avec tous les risques que cela comporte. On peut citer le cas récent de Mitnick qui a réussi impunément pendant des années à piller le réseau internet pour des millions de dollars. Son dernier fait d'arme fut de casser la sécurité du super-ordinateur de San Diego, l'ordinateur le mieux protégé au monde. Il faut donc rester très prudent en matière de sécurité et ne pas faire une confiance aveugle aux procédures de sécurité mises en place. Au contraire, il faut évaluer les risques et les coûts qu'occasionnerait une brèche dans la sécurité du système.

Deuxièmement, si on pense pouvoir un jour arriver à une sécurité quasi-absolue dans les échanges de données électroniques, elle ne serait pas à mettre dans toutes les mains et ceci pour des raisons politiques et judiciaires. Les gouvernants des états n'auraient plus aucun contrôle sur ce qui entre ou sort de leur pays, en particulier au sujet des transferts financiers. C'est ce qui a déjà pris place en ex-Urss où la mafia locale, qui est bien équipée en technologie de pointe, peut réaliser ses trafics financiers sans aucun moyen de contrôle de la part des gouvernements et services anti-fraudes concernés. Ainsi, elle réalise de nombreux achats de société dans le monde entier et blanchit par la même occasion son argent grâce à de simples échanges de données électroniques auxquelles on aurait appliqué le service de confidentialité.

Ce mémoire ne traitant pas de « Technology Assessment », on ne fera plus référence aux multiples implications politiques inhérentes à l'introduction de la sécurité dans les échanges de données électroniques, on se limitera à analyser les aspects techniques de l'intégration de la sécurité dans le standard EDIFACT.

PARTIE II :

ELEMENTS DE SECURITE

Les architectures de sécurité OSI font une distinction claire entre les *services* de sécurité, les *mécanismes* de sécurité et les *protocoles*. Par exemple, le chiffrement est un mécanisme qui peut être utilisé comme partie intégrante de nombreux services dont le plus évident est le service de confidentialité. Le protocole est la façon d'utiliser un ensemble d'outils (mécanismes) pour garantir un service de sécurité. Cette partie II sera décomposée en trois chapitres selon la vue logique énoncée ci-dessus : Les services de sécurité au chapitre 3, les mécanismes au chapitre 4 et les protocoles au chapitre 5.

Chapitre 3

Services de sécurité pertinents

Dans le chapitre 1, au point 2.2, sont exposées les questions clefs pour évaluer l'étendue des risques auxquels s'expose une organisation lorsqu'elle utilise le système EDI qu'on lui propose (ou impose). Certaines de ces questions, plus spécifiquement celles qui traitent des dangers menaçant un message EDI en transit, vont trouver une réponse dans ce chapitre. Ces solutions seront formalisées sous forme de quelques services de sécurité; ces derniers furent d'ailleurs développés antérieurement à nos considérations pour assurer la sécurité des réseaux informatiques. Pour l'ISO, les services de sécurité sont classés en cinq groupes : **la confidentialité**, **l'authentification**, **l'intégrité**, **la non-répudiation** et **le contrôle d'accès**. Notons, cependant, que le programme TEDIS II [TED2, 92] propose d'aller plus loin et examine de nouveaux services de sécurité plus spécifiques à l'administration et au commerce.

Dans ce chapitre sont présentés les différents services de sécurité pour échanger de manière sûre des messages EDIFACT. Chaque service décrit un aspect du système de sécurité vu par l'utilisateur et est donc conçu pour satisfaire les demandes naturelles de l'utilisateur. Pour rencontrer les exigences de ces utilisateurs de l'EDI, on procède habituellement comme suit : pour chaque document commercial ou administratif, on envoie aux intéressés un questionnaire les interrogeant sur la pertinence de chaque service de sécurité qu'on pourrait lui appliquer. Pour chacun des cinq services, on donnera une appréciation globale, et non pas document par document, de sa pertinence en guise de conclusion.

Pour illustrer chacun des services, l'on présente une figure démontrant les dangers encourus si on ne fournit pas aux utilisateurs abusés (l'émetteur, le destinataire ou les deux) le service décrit. Notons que seul le contenu informatif (logique) des messages véhiculés entre les partenaires commerciaux y est présenté (et non pas des messages sous format EDIFACT). La raison en est que le service est le même quelle que soit la présentation qui est faite aux données; d'ailleurs, on retrouve ces différents services dans les messageries électroniques de type X.400, ou du moins des services apparentés.

Le terme authentification est utilisé dans deux contextes différents :
→L'**authentification de document** est le mécanisme qui permet de s'assurer qu'un document a bien été émis par une personne autorisée (authentification de l'origine) et qu'il n'a subi aucune modification illicite (authentification du contenu).
→L'**authentification d'un acteur** consiste à s'assurer de la « véritable » identité de l'acteur (humain ou processus).

Dans la suite, on utilisera aussi bien les définitions ci-dessus que les termes OSI pour décrire les cinq services de sécurité, les correspondances suivantes permettent de faire le lien entre ces différentes terminologies :

→authentification ≅ authentification de l'origine,
→intégrité ≅ authentification du contenu et
→contrôle d'accès ≅ authentification d'un acteur.

Toute communication dans un environnement potentiellement non sûr devrait commencer par établir l'identité des parties communicantes. Il est naturel de commencer par le service d'authentification des acteurs.

1 L'authentification des acteurs

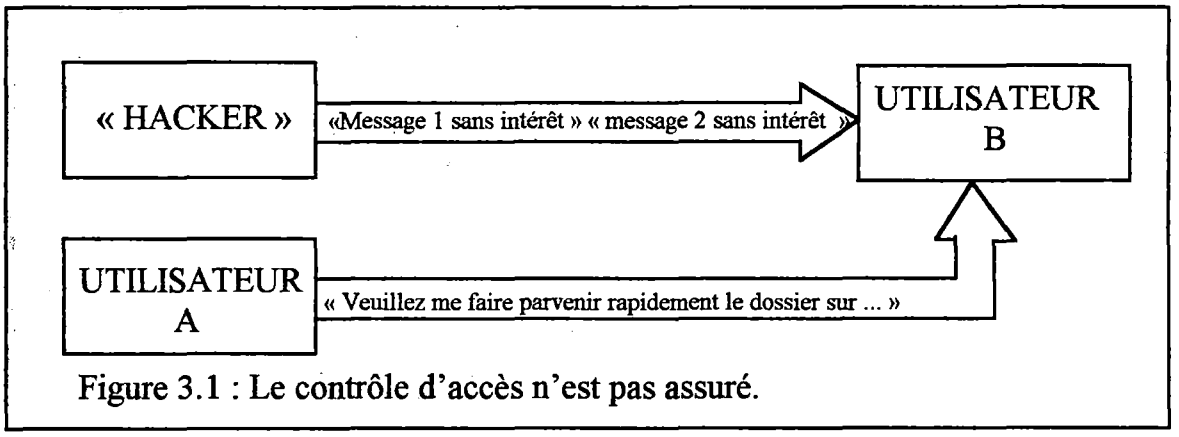

Figure 3.1 : Le contrôle d'accès n'est pas assuré.

Pour pouvoir envisager la sécurisation d'un système, la première chose qu'il faut être capable de faire avant tout, c'est identifier correctement l'acteur qui veut utiliser le système. Il faut ainsi pouvoir vérifier si une personne qui se présente au système (au terminal p.ex.) est bien la personne qu'elle prétend être. Cette première forme d'authentification d'acteur est ce qu'on pourrait appeler « le **contrôle d'accès** ».

Le contrôle d'accès est nécessaire dans le but de protéger un réseau contre la saturation délibérée d'un opposant (cas I) mais normalement il protège un service informatique contre les accès non-autorisés (cas II).

Dans le monde EDI, on peut citer l'exemple suivant pour le cas II : les personnes autorisées à émettre des messages de l'organisation à laquelle ils appartiennent doivent empêcher des personnes non-autorisées d'envoyer des messages depuis leurs propres systèmes. L'envoi d'un message incorrect ou frauduleux de la part d'un jeune membre de l'équipe en utilisant le log-on ID et le mot-de-passe du manager, par exemple, peut faire entrer l'organisation dans un contrat qu'elle ne pourra pas respecter.

La figure 3.1 illustre le cas I où un émetteur non-autorisé, que l'on a nommé « hacker » harcèle l'utilisateur B avec des messages sans intérêt, peut-être dans le but de freiner ses activités ou du moins ralentir ses échanges EDI, en saturant les lignes qui donnent accès à l'utilisateur B.

Ce service primordial est souvent le premier à être intégré dans toutes applications faisant usage d'un réseau. C'est au niveau du logiciel EDI que ce service est implémenté. Il est d'ailleurs impensable logiquement d'intégrer ce service dans un standard de représentation de données. Le logiciel EDI devra réaliser au moins deux niveaux de vérification sécuritaire : sur l'opérateur qui donne des commandes au système et sur toutes communications avec le réseau ou un système EDI externe. En ce qui concerne l'opérateur au terminal, le contrôle doit être physique et logique. On ne voudrait pas que quelqu'un puisse s'asseoir au terminal et tape une commande du type « del *.* » . Il est tout aussi important qu'il ne puisse pas être capable de faire une commande pour, disons, une douzaine de bateaux de guerre ou 5 tonnes d'haricots.

Les mécanismes qui apportent des solutions au contrôle d'accès sont très connus. On peut citer le PIN (Personal Identification Number) et le mot de passse qui sont certainement les moyens les plus répandus. On dispose également d'autres techniques ou technologies tels les « smart cards » (cartes à microprocesseur), le « token » (jeton) ou, encore, les techniques biométriques (les empreintes digitales, la reconnaissance vocale ainsi que la reconnaissance de la rétine de l'oeil).

Une autre forme d'authentification d'acteur prend place quand un utilisateur B veut se convaincre qu'il communique bien avec A, un autre utilisateur du système. L'**identification de l'utilisateur** est un processus consistant toujours en une première étape où B est capable de vérifier que l'utilisateur A est un utilisateur existant dans le système, enregistré avec un ID-number, avec certaines caractéristiques tels les privilèges utilisateurs, etc..., et également avec une clé publique.

Il peut y avoir une seconde étape interactive où B échange des messages avec l'utilisateur en question, après quoi il sera convaincu que l'utilisateur est en réalité A.

La première étape est pertinente dans à peu près tous les systèmes de sécurité, alors que la seconde n'est pas pertinente dans les systèmes basés sur EDIFACT, puisqu'un tel système ne permet pas souvent d'interaction, ou du moins il le permet seulement avec un délai de temps significatif.

Pour bien comprendre ce service et pour illustrer les deux étapes qui nous sont proposées ci-dessus, nous allons faire une petite analogie avec une communication téléphonique. Dans 99% des cas, on n'imagine pas tenir une conversation téléphonique alors que l'on ne sait pas avec qui on la tient. Si les deux interlocuteurs se connaissaient relativement bien préalablement à l'appel, ils peuvent s'identifier au son de leurs voix (dès le mot protocolaire « allo »). Plaçons-nous alors dans l'autre hypothèse, qui est plus adéquate à notre propos vu que l'on ambitionne l'open EDI, une forme d'EDI où les partenaires commerciaux ne se connaissent pas. Pour identifier son interlocuteur, on lui demande de décliner son identité, son adresse et son numéro de téléphone, par exemple. La première étape du processus consiste à vérifier que cette personne existe réellement, ici, cela peut se faire en consultant un botin téléphonique où nom, adresse et numéro de téléphone doivent coïncider; cependant, rien ne prouve que c'est elle qui nous parle! Par contre, la deuxième étape du processus permet d'établir, sur base d'échange de messages, que son interlocuteur est celui qui se revendique être depuis la première étape. On a développé, en cryptologie, des protocoles appelés « challenge/response protocol ». Pour une communication

téléphonique, un challenge serait de demander à son interlocuteur des informations qu'il doit absolument connaître et qu'ignorent, si possible, les autres utilisateurs. Par exemple, leur précédente conversation téléphonique. On peut également imaginer une authentification mutuelle d'acteurs, il faut alors, au minimum, échanger trois messages.

Ce service est aussi capital que le contrôle d'accès. L'absence de la deuxième étape du processus semble être un handicap à l'établissement complet du service d'identification de l'utilisateur. En fait, ce n'est pas trop grave car la présence de la clé publique dans les caractéristiques d'identification permettra a posteriori de vérifier « l'honnêteté » du partenaire, et ceci grâce aux autres services de sécurité qui en feront usage.

Bien sûr, le contrôle d'accès et l'identification des utilisateurs ne sont souvent pas suffisants. Qu'est-ce qui nous prouve qu'un document portant le nom d'un utilisateur A provienne effectivement de cet utilisateur A? Le service d'authentification de l'origine apporte une réponse à cette interrogation. Pour faciliter la compréhension des dessins, l'utilisateur qui tente de frauder s'appellera toujours H comme Hacker.

2 Authentification de l'origine

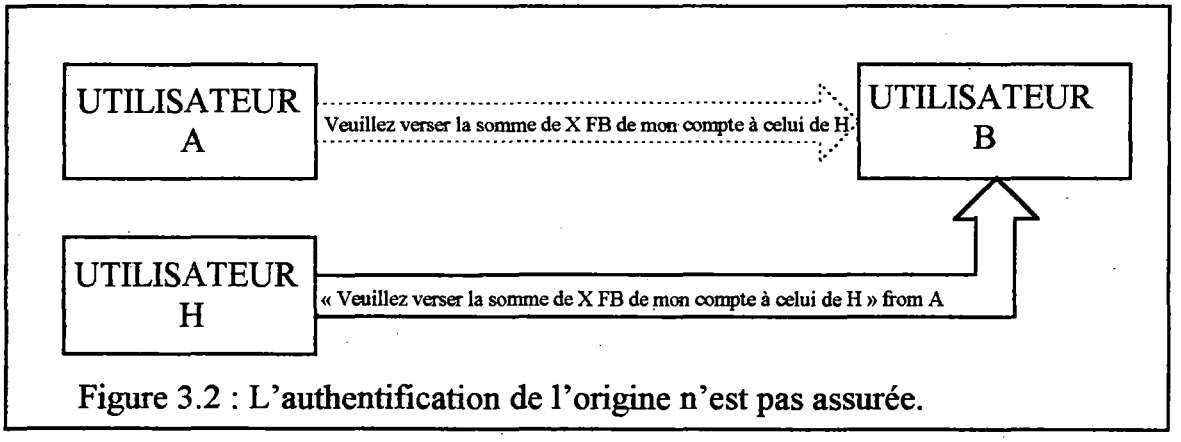

Figure 3.2 : L'authentification de l'origine n'est pas assurée.

Ce service est apparenté à la classe des services d'authentification du monde OSI, plus particulièrement, l'authentification de l'origine des données dont la définition est : « corroboration que la source des données reçues est celle qui est revendiquée ».

Lors d'une communication entre deux utilisateurs A et B, un troisième utilisateur H pourrait envoyer un message sous une fausse identité, celle de A, et tromper ainsi le destinataire B sur la véritable identité de l'émetteur du message. Le service d'authentification fera en sorte que B puisse vérifier que le message provient réellement de A.

Dans le monde EDI, qu'est-ce qui nous assure que le message reçu est de l'organisation déclarée? Si le service d'authentification de l'origine n'est pas établi, alors une entreprise compétitrice peut utiliser l'EDI pour obtenir, par exemple, les

listes de prix ou des informations sensibles sur le planning de production. Une banque peut agir sur instructions qui apparaissent provenir d'un client mais qui sont en fait l'oeuvre d'un brigand.

La figure 3.2 illustre cette possibilité de fraude. L'utilisateur B, qui est une banque, reçoit un ordre de paiement avec l'en-tête du client A et doit réaliser un transfert d'argent du compte de ce même client (A) vers le compte de H. Ce message ne provient pas d'une communication entre A et B, c'est pour cela qu'il est dessiné dans un canal de communication en pointillé. En réalité, c'est un message de H reprenant frauduleusement l'en-tête de A. La banque réalise la transaction sans se douter que celle-ci se fait dans le dos de A.

En conclusion, le service d'authentification de l'origine est un service essentiel qui est exigé par les utilisateurs d'EDI pour pratiquement tous les documents commerciaux et administratifs.

Une autre demande assez naturelle pour avoir une communication sûre est que les messages ne pourront pas être changés par une tierce partie sans qu'il y ait détection de la fraude. Ceci signifie qu'il faut assurer l'intégrité du contenu des messages échangés.

3 Intégrité du contenu

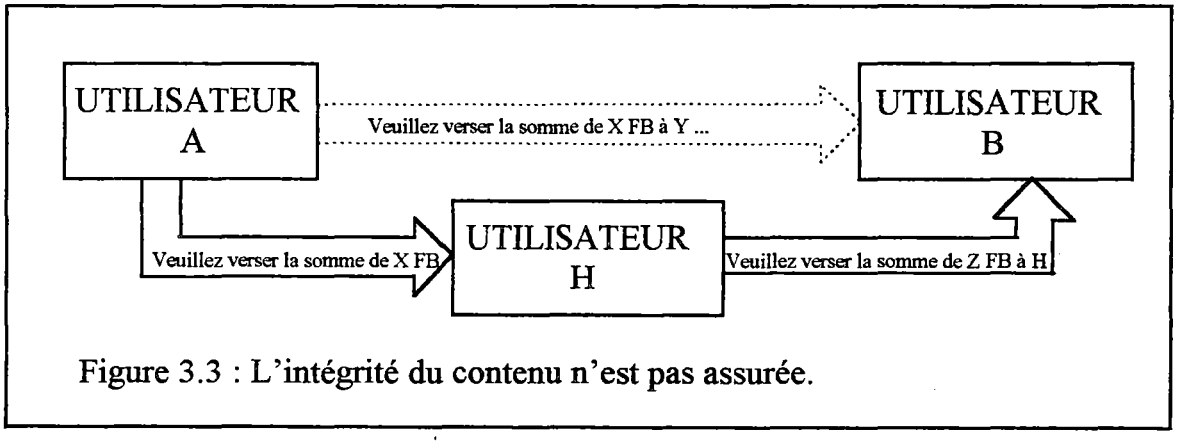

Figure 3.3 : L'intégrité du contenu n'est pas assurée.

Ce service d'intégrité du contenu est décrit par ISO en ces termes : « est la propriété que les données n'ont pas été altérées ou détruites d'une manière non-autorisée ».

B reçoit un message de A. B peut vérifier que le message n'a pas été changé par une tierce personne H sur le chemin. En d'autres termes, il est impossible qu'une tierce partie surgisse avec un message que B va accepter et qui n'a jamais été envoyé sous cette forme par A.

En pratique, l'intégrité du contenu et l'authentification de l'origine des données sont liées de deux façons; elles emploient habituellement les mêmes mécanismes et

elles sont habituellement demandées ensemble. En ce qui concerne ce besoin couplé, assurer que les données reçues viennent de la source revendiquée n'est pas utile en pratique si les données peuvent avoir été changées d'une façon non-autorisée. Inversement, savoir que les données ont été transmises sans aucun changement non-autorisé n'est pas utile à moins de savoir que les données viennent de la source revendiquée et pas d'un imposteur.

La figure 3.3 reprend l'exemple de l'ordre de paiement. A envoie à sa banque un message demandant de transférer de l'argent de son compte vers celui de l'utilisateur Y. Supposons, de plus, que A ait fait le nécessaire pour que la banque puisse authentifier l'origine du message. Si le service d'intégrité du contenu n'est pas assurée, le scénario suivant risque d'avoir lieu. Un utilisateur H intercepte le message de A. Par conséquent, ce message est dessiné en pointillé dans un canal de communication entre A et B, puisqu'il n'arrivera pas sous la forme originelle que lui avait donnée A. L'utilisateur H modifie à son avantage le message de A, par exemple, il met son nom en lieu et place du véritable destinataire de la transaction financière. Ensuite, H envoie le message à l'utilisateur B. L'utilisateur B, une banque en l'occurrence dans notre exemple, reçoit le message de H. Cependant, la banque sera convaincue qu'il s'agit d'un message de A, un client de la banque, puisque le service d'authentification de l'origine est assurée, elle versera la somme du compte de A à celui de H. En définitive, H a réussi à détourner à son avantage de l'argent.

En conclusion, le service d'intégrité du contenu est un service tout aussi important que le précédent, il est pertinent aux yeux des utilisateurs pour à peu près tous les documents commerciaux et administratifs.

Ces deux premiers services protègent A et B vis-à-vis des autres utilisateurs lors de leurs communications mais ils ne protègent pas B contre la mauvaise foi de A (cas I), ou inversement, A contre la mauvaise foi de B (cas II). Ces deux cas de figure sont résolus par respectivement la non-répudiation de l'origine et la non-répudiation de la réception. La bonne compréhension de ce qui vient d'être dit demande quelques explications. Plaçons-nous dans l'hypothèse que les services d'authenticité de l'origine et d'intégrité du contenu sont assurés.

Pour le cas I, B reçoit un message de A, B peut vérifier que l'émetteur est A (a) et que le contenu n'a pas été altéré (b), cependant , dans un cas de litige où A nie avoir envoyé le message, B ne pourra pas convaincre une tierce partie de ces deux réalités (a et b) vu que rien n'interdit B de construire lui-même un tel message.

Pour le cas II, même si B reçoit le message de A, B peut ensuite nier l'avoir reçu, une attitude facile à défendre vu qu'un message peut se perdre dans le réseau et ne jamais arriver à son destinataire. D'un autre côté, A n'a aucune preuve que B ait reçu le message.

Ces deux problèmes (I et II) sont résolus respectivement par la non-répudiation de l'origine et de la réception.

4 Non répudiation de l'origine

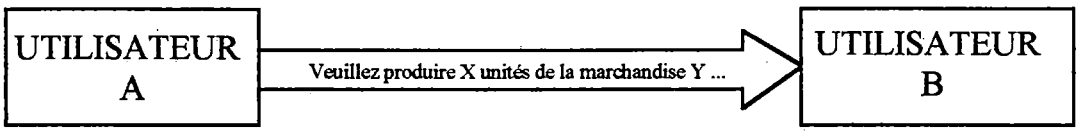

Figure 3.4 : Répudiation de l'origine.

La définition donnée par OSI pour la répudiation est : « le fait qu'une entité impliquée dans une communication nie avoir participé à une partie ou à toute la communication ». Le service de non-répudiation peut prendre deux formes : la non-répudiation de l'origine et la non-répudiation de la livraison. Ce point 4 explicite le premier des deux services.

B reçoit un message de A ainsi qu'une preuve que le message provient effectivement de A. L'utilisateur B ainsi que tous les autres utilisateurs pourront vérifier cette preuve. Pour tout utilisateur différent de A, il est impossible de fournir une preuve que le message provient de A à moins que A lui-même ait envoyé une preuve pour ce message.

La figure 3.4 illustre ce service. Le fait de savoir que la partie émettrice du message ne pourra pas, à une date ultérieure, renier la connaissance de ce message ou de son contenu est important. Par exemple, si une organisation (utilisateur B sur le dessin) fabrique et expédie des marchandises en conformité avec le message reçu de la part de son client (utilisateur A sur le dessin) et que, par la suite, le client nie la connaissance de la commande, alors le fournisseur (utilisateur B sur le dessin) devra supporter les coûts de fabrication de ces marchandises.

La pertinence des deux services de non-répudiation sera discutée dans la conclusion du point suivant (5).

Un message qui intime un ordre à son destinataire est un exemple où le destinataire du message ne devra pas, plus tard, être en mesure de nier qu'il l'a reçu. Cet exemple motive le service de non-répudiation de la réception que l'on va maintenant décrire.

5 Non répudiation de la réception

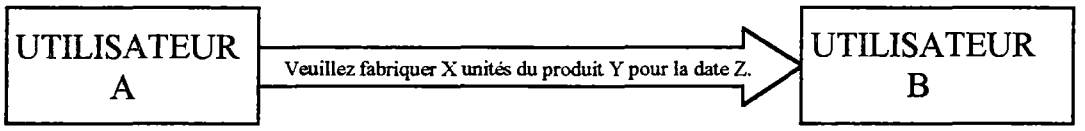

Figure 3.5 : Répudiation de la livraison.

B reçoit un message de A et envoie en retour un accusé de réception à B, qui peut être vérifié comme étant valide par qui que ce soit (incluant B bien sûr). Seul B est capable de produire un accusé de réception valide et par conséquent B ne pourra pas plus tard nier avoir reçu le message.

La figure 3.5 illustre ce service sur base d'un message analogue à la figure 3.4. Mais dans ce cas de figure, A ne sera jamais livré pour sa commande de marchandises. Une raison pourrait être que le fournisseur n'ait pas pu respecter le délai de livraison (date Z). A va s'enquérir d'une explication auprès de B mais celui-ci peut lui répondre qu'il n'a pas reçu le message si jamais le service de non-répudiation de la livraison n'a pas été réclamé par A.

En conclusion, ces services protègent l'émetteur vis-à-vis du destinataire et vice versa, la pertinence de ces services dépendra du niveau de confiance mutuelle des partenaires commerciaux. Puisque l'on veut mettre en place des échanges commerciaux, via l'EDI et le standard EDIFACT, ouverts au plus grand nombre d'utilisateurs possibles (Open EDI), on sera en présence de partenaires commerciaux qui se connaissent peu ou prou. Dans cette optique, les services de non-répudiation revêtent une importance considérable.

6 Confidentialité du contenu

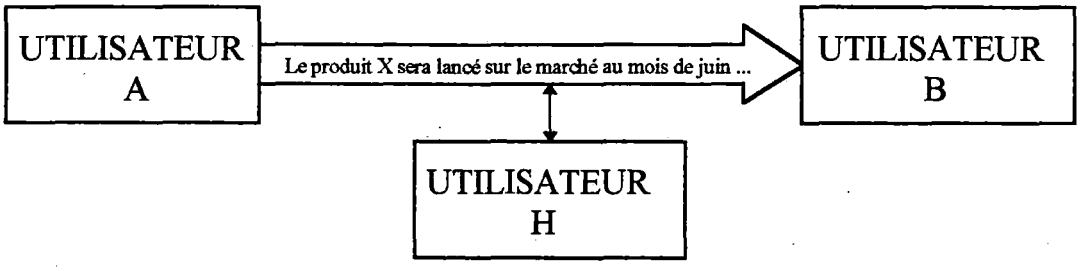

Figure 3.6 : La confidentialité du contenu n'est pas assurée.

La définition qui nous est donnée par OSI est que la confidentialité des données assure que l'information n'est pas disponible ou dévoilée à des individus, entités ou processus non-autorisés.

B reçoit un message de A. Le message est transporté de telle façon qu'un parti tiers H ne pourra rien apprendre sur le contenu du message.

Sur la figure 3.6, le service n'est pas implémenté et l'utilisateur H peut intercepter le message, le lire puis le réexpédier sur le réseau. Ce service apparaît nécessaire chaque fois que l'on désire envoyer des données confidentielles, par exemple, des données sur la vie privée des gens ou des données exposées à l'espionnage industriel.

En conclusion, le service de confidentialité n'est pas pertinent pour beaucoup de documents.

Chapitre 4

Primitives de sécurité

Au chapitre précédent, on a identifié cinq services de sécurité pertinents pour les échanges de messages EDI. Ce chapitre prend en charge les primitives de sécurité requises pour ces cinq services. On va donc faire appel à quelques-uns des mécanismes de sécurité qui ont été développés dans le domaine de la sécurité. De part la nature électronique des documents échangés lors de transfert de messages EDI, les mécanismes les plus importants sont tous issus de la cryptologie; celle-ci étudie les techniques de chiffrement. Les techniques de chiffrement sont de deux types : les méthodes conventionnelles ou cryptosystèmes symétriques et les méthodes à clé publique ou cryptosystèmes asymétriques. On se penchera plus particulièrement et plus en détail sur un mécanisme dérivé du chiffrement à clé publique : **la signature digitale**. L'idée fondamentale de ce mémoire est d'ajouter des signatures digitales aux messages EDI.

1 Chiffrement

Le principe de base pour le chiffrement est dessiné dans la figure 4.1. Les termes rencontrés dans la figure 4.1 font référence aux notations suivantes :

Soient - M : le message écrit en clair par l'émetteur (Plaintext).
- M' : le texte chiffré (Ciphertext).
- E : l'algorithme de chiffrement.
- D : l'algorithme de déchiffrement.
- k : une clé.

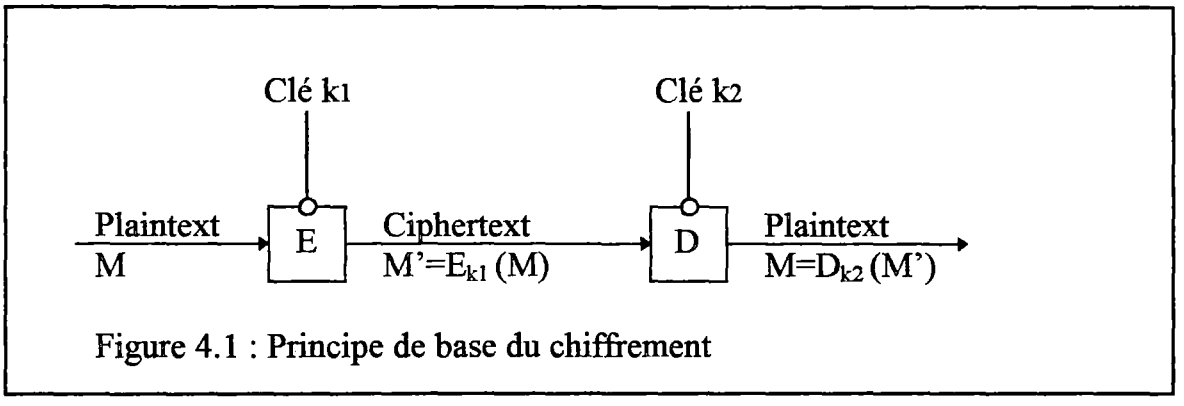

Figure 4.1 : Principe de base du chiffrement

Le mécanisme du chiffrement s'exprime simplement par : $M' = E_{k1}(M)$ où k1 est une clé de chiffrement. Le texte chiffré s'obtient par l'application de la fonction E_{k1} sur le texte en clair. Cette fonction a comme paramètre le texte d'origine et la clé, donc on aurait pu aussi bien écrire E(k1, M) que $E_{k1}(M)$. Il faut également se doter du mécanisme inverse, le mécanisme de déchiffrement qui permet de retrouver le texte en clair à partir du texte chiffré : $M = D_{k2}(M')$ où k2 est une clé de déchiffrement.

Les techniques de chiffrement requièrent de bons algorithmes cryptographiques mais également une bonne gestion des clés (key management) pour garantir la sécurité d'un système EDI qui en ferait usage. De nombreux algorithmes standards existent aussi bien pour le chiffrement à clé secrète que publique; on se reportera utilement à la littérature pour une explication détaillée du fonctionnement de chacun des algorithmes de sécurité rencontrés dans ce mémoire [1]. De son côté, la gestion des clés - qui va de la génération des clés, la distribution et le stockage des clés, l'enregistrement et la certification des clés (uniquement pour les systèmes à clé publique) à la destruction des clés et leurs remplacements - ne sera pas détaillée dans ce mémoire. La raison en est que la gestion des clés n'est pas nécessaire pour décrire l'intégration formelle de la sécurité dans EDIFACT. Mais, d'un autre côté, elle est vitale si l'on veut implémenter un système EDI sécurisé.

2 Méthodes conventionnelles

2.1 Chiffrement à clé secrète

En gardant les mêmes conventions qu'au point 1, la figure 4.2 schématise les opérations de chiffrement et de déchiffrement mise en place dans le chiffrement à clé secrète.

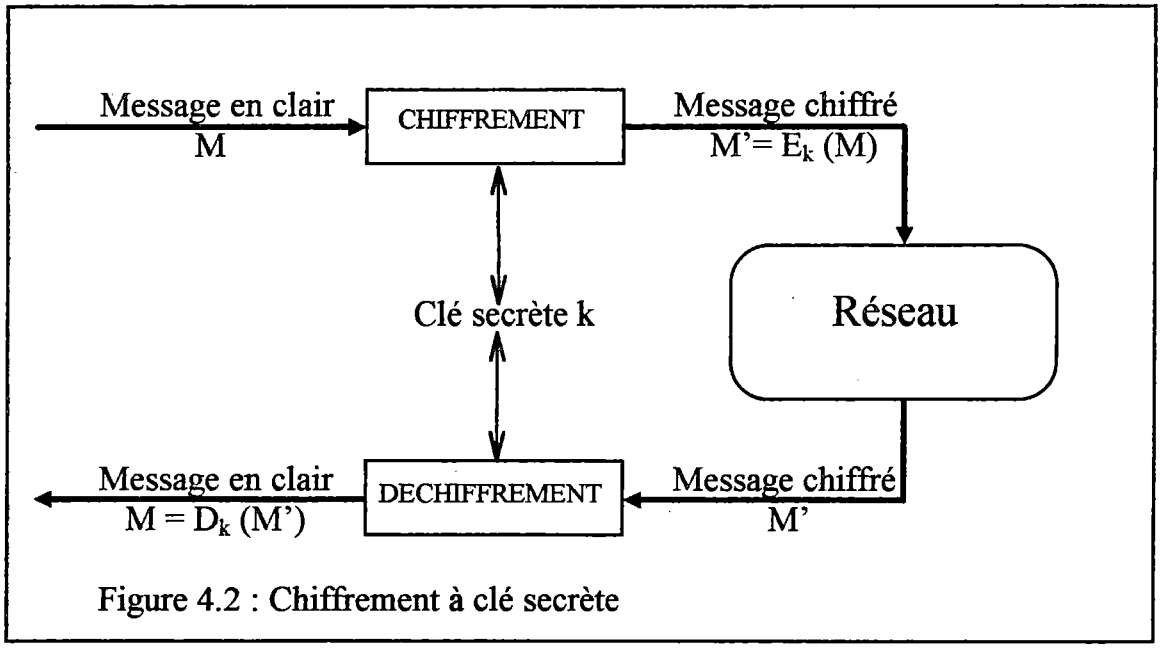

Figure 4.2 : Chiffrement à clé secrète

Les deux partenaires qui s'échangent mutuellement des messages partagent un secret : une clé secrète k commune. Les algorithmes de chiffrement et de déchiffrement, notés respectivement E et D, peuvent être connus de tous; ce sont des algorithmes publics. Par contre, les fonctions de chiffrement et de déchiffrement, notés respectivement E_k et D_k, sont connues uniquement par ceux qui connaissent k.

On peut résumer les opérations de chiffrement et de déchiffrement comme ceci : $M' = E_k (M)$ et $M = D_k(M') = D_k (E_k (M))$. Ce chiffrement est parfois appelé chiffrement symètrique car la connaissance de la clé par chacun permet la communication dans les deux sens. Il porte également le nom de système à clé privée.

Cette méthode a la propriété suivante : aucune tierce partie ne peut trouver M à partir de E_k (M) sans connaître la clé k.

L'algorithme de chiffrement à clé secrète le plus utilisé est le DES (Data Encryption Standard). Le standard DES a été développé par IBM pour le département de la défense des Etats-Unis et fut subséquemment publié en tant que standard.

Cette méthode n'est pas exempte de reproche, on peut en citer trois :

- Les deux partenaires doivent posséder la même clé, ce qui implique qu'ils ont dû se l'échanger auparavant. Cet échange a dû lui-même demandé des conditions parfaites de sécurité.
- Chaque partenaire gère la sécurité de sa clé secrète avec une rigueur qui lui est propre. Or, l'appropriation de cette clé par un « hacker » peut se révéler très vite catastrophique pour les deux partenaires commerciaux. Cette dépendance mutuelle peut se révéler dangereuse, particulièrement si on imagine que les partenaires ne se connaissent pas entre eux. Un paliatif est de réduire la durée de vie de cette clé, par exemple, en la limitant à une seule communication.
- Soit on partage la même clé avec tous ses partenaires commerciaux, mais alors on s'expose au problème de la mascarade. La mascarade est une situation où une personne se fait passer pour une autre, ce qui est aisé dans ce cas-ci puisqu'ils sont tous en possession de cette clé secrète. Soit on dispose d'une clé secrète par partenaire commercial, ce qui entraîne que chaque partenaire devra gérer N-1 clés et que le système comptera N (N-1) / 2 clés au total.

Les méthodes à clé publique ne présente pas ces inconvénients, qui sont de taille dans notre optique de travail. Car le but ultime de l'intégration de la sécurité dans EDI est d'atteindre un EDI ouvert à tous (open EDI), c'est-à-dire un monde EDI où les partenaires commerciaux ne se connaissent pas nécessairement mais où tout sera mis en oeuvre pour qu'ils puissent cependant commercer de façon sécurisée entre eux. Et cela pour le plus grand nombre possible de partenaires commerciaux. Or, le chiffrement à clé secrète nécessite un échange préalable de clés secrètes, repose sur la confiance mutuelle des entités communiquantes et demande la gestion d'un nombre de clé secrète qui augmente proportionnellement au nombre de partenaires; il devra donc être complété par d'autres techniques. Pour atteindre notre but, on pense plutôt à la signature digitale, un mécanisme basé sur le chiffrement à clé publique et qui sera étudié au point 4.

2.2 Manipulation Detection Code (MDC)

Pour un message M, un code de détection de manipulation MDC (M) est calculé. Le message peut avoir n'importe quelle longueur mais le MDC (M) a une longueur fixe relativement courte, qu'on appelle un condensé.

Et ce MDC (M) a la propriété suivante, que l'on appelle l'« absence de collision » : il est impossible de trouver deux messages différents M1 et M2 tel que MDC (M1) = MDC (M2). Cette fonction MDC est souvent désignée sous le nom de « hash function ». Les recommandations ISO DP 10118 décrivent des méthodes pour calculer les MDCs.

Un condensé (« hash value ») d'un document électronique, c'est-à-dire d'un string de bits, est typiquement un string de bits plus court calculé à partir du string originel par une méthode rendue publique. Pour des raisons de vitesse et de place, le condensé (« hash

value ») devra être le plus court possible, alors que, d'un autre côté, il devra être suffisamment long pour éviter les collisions avec une très grande probabilité.

2.3 Message Authentification Code (MAC)

A partir d'un message M et d'une clé secrète k, on calcule un code d'authentification du message MAC_k (M). Le message peut avoir n'importe quel longueur mais le MAC a une longueur fixe relativement courte. Le MAC est un condensé cryptographique du message.

Et ce MAC_k (M) a la propriété suivante : même après avoir vu plusieurs paires de messages et de MAC correspondants, mais sans connaître la clé k, il sera impossible de trouver un message M1 et un MAC X tel que le calcul de MAC_k (M1) nous donne la valeur X. On implémentera le MAC selon les recommandations de l'ISO DIS 9797.

3 Méthodes à clé publique

3.1 Chiffrement à clé publique

En gardant les mêmes conventions qu'au point 1, la figure 4.3 schématise les opérations de chiffrement et de déchiffrement mise en place dans le chiffrement à clé publique.

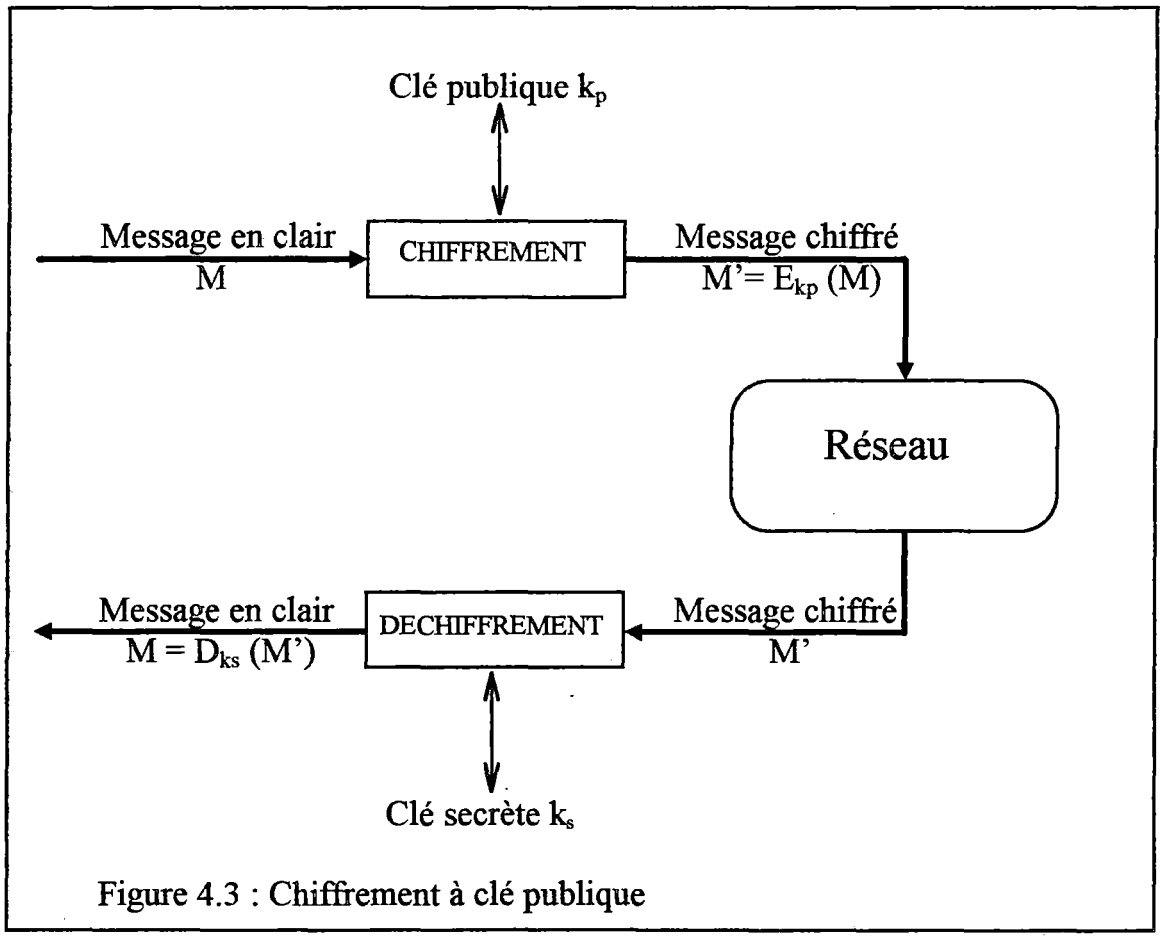

Figure 4.3 : Chiffrement à clé publique

Les deux partenaires n'ont plus en commun la connaissance de la clé secrète et, par conséquent, l'émetteur et le récepteur ne pourront plus se substituer l'un à l'autre, le chiffrement est alors asymétrique. Le récepteur R possède une clé secrète S_k, connue de lui seul. Il existe une clé publique correspondante P_k, qui est connue de tous les utilisateurs. Parmi tous ces utilisateurs qui connaissent P_k et qui peuvent émettre un message vers R, choisissons en un qui devient l'émetteur. L'émetteur peut, à partir d'un message M et en utilisant la clé publique P_k, calculer le chiffrement E_{Pk} (M) et l'envoyer à R. En utilisant sa connaissance de S_k, R peut déchiffrer le message et obtenir $M = D_{Sk}$ (E_{Pk} (M)).

Cette méthode a la propriété suivante : même en connaissant P_k, mais sans connaître S_k, il est impossible pour une tierce entité de trouver M à partir de E_{Pk} (M).

Le point délicat est donc de générer deux clés P_k et S_k liées entre elles de telle manière que l'utilisateur de la clé P_k ne puisse en déduire S_k mais dont P_k a pu être construite à partir de S_k! L'algorithme par excellence qui entre dans cette catégorie est le RSA (Rivest, Shamir et Adelman).

Cette méthode n'est pas sans inconvénients non plus, le plus critique est le suivant :

• Ce type de chiffrement est basé sur le calcul de fonctions telle la mise en puissance de nombres de plus de 50 chiffres. Il demande un équipement hardware assez performant pour éviter des temps CPU trop coûteux.

Cet inconvénient majeur nous incitera, chaque fois que cela n'handicapera pas de trop le service de sécurité, à combiner les techniques à clé publique avec les techniques à clé secrète de manière à accélérer les calculs.

Par contre, le chiffrement à clé publique présente entre autre l'avantage suivant sur le chiffrement à clé secrète :

• il simplifie grandement la gestion des clés comparé aux systèmes à clé privée classique. Pour un système de n utilisateurs, le chiffrement à clé publique demande seulement n paires de clé (secrète et publique), alors que n (n-1) / 2 clés secrètes sont nécessaires dans un système à clé privée. En outre, chaque utilisateur devra stocker seulement une clé secrète et n clés publiques, comparé aux n-1 clés secrètes qu'il faut stocker pour chaque utilisateur dans un cryptosystème à clé privée.

Sur base de cette méthode générique, on a développé deux techniques dont on fera largement usage dans la suite : la signature digitale et le certificat.

4 Signature digitale.

4.1 Introduction

En vue de faire du commerce par l'intermédiaire de documents électroniques, il est nécessaire de remplacer toutes les procédures qui permettaient de sécuriser les documents sur papier par d'autres procédures de nature électronique mais qui soient tout aussi efficaces dans le but qu'elles obtiennent également un statut juridique, comme le soulignait le point 2.1 du premier chapitre.

Le but premier des documents est de convoyer ou préserver de l'information. Des siècles durant, la façon la plus pratique de produire des documents a été de les

écrire ou de les imprimer sur un bout de papier. Si l'intégrité de ses papiers doit être assurée, des mesures spéciales doivent être prises de telle façon qu'il sera difficile de faire un changement quelconque au contenu des documents. Ces mesures peuvent inclure par exemple l'utilisation de papiers spéciaux. Mais l'une des mesures les plus habituelles est d'ajouter une signature manuscrite au document. Celle-ci identifie le document original et le rattache à un individu ou à des individus. La signature peut, de plus, indiquer l'accord avec ce qui a été laissé sur le document.

D'un autre côté, les documents signés sont considérés comme essentiels pour les aspects légaux. Aller plus loin dans l'open EDI et ne rien faire pour améliorer la sécurité des messages EDI serait un peu comme ne jamais signer une lettre d'affaire, ne jamais signer un chèque, ne jamais signer un contrat.

Il faut inventer une technique qui serait l'équivalent digital de la signature manuscrite : la « signature digitale ». Lorsque le concept de signature digitale fut, pour la première fois, discuté par Diffie et Hellman, ils suggérèrent une solution au problème en utilisant des cryptosystèmes à clé publique.

Leur caractéristique principale est qu'ils séparent le chiffrement du déchiffrement de telle façon que (1) une personne peut chiffrer des messages de telle façon que beaucoup de gens pourront les lire ou (2) beaucoup de gens pourront chiffrer des messages de telle façon qu'une seule personne pourra les lire. La deuxième option (2) donne lieu au chiffrement à clé publique. C'est la première option (1) qui permet de signer un message purement digital. Les algorithmes utilisés dans ces cryptosystèmes sont basés sur des « trapdoor one-way functions », qui signifie que seule une personne en possession de l'information secrète pourra retourner en arrière et calculer l'inverse de la fonction.

4.2 Description technique

Le mécanisme de signature digitale est très proche de celui du chiffrement à clé publique à la différence près que l'ordre des clés est inversé : le message est cette fois chiffré (on dira que le message est signé) avec la clé secrète : $M' = E_{ks}(M)$ et le déchiffrement (on parlera de vérification de la signature) est réalisé grâce à la clé publique : $D_{kp}(M') = M$. La figure 4.4 propose une illustration de ce mécanisme.

Un utilisateur peut signer un message en le chiffrant avec sa propre clé secrète. Toute personne ayant accès à la clé publique pourra vérifier en le déchiffrant que celui-ci a été signé par la clé secrète correspondante, mais cela ne l'aidera pas à créer un autre message avec cette caractéristique. Ceci nous amène aux deux propriétés essentielles des signatures digitales :

- ➜ le récepteur d'un message ne peut apposer une fausse signature.
- ➜ un signataire ne peut pas désavouer une signature.

Cette méthode permet donc à un récepteur d'un message purement électronique qui posséde une signature digitale de démontrer à d'autres personnes qu'il provient d'une personne particulière, exactement comme une signature manuscrite sur une lettre permet au destinataire de rattacher le contenu à son auteur.

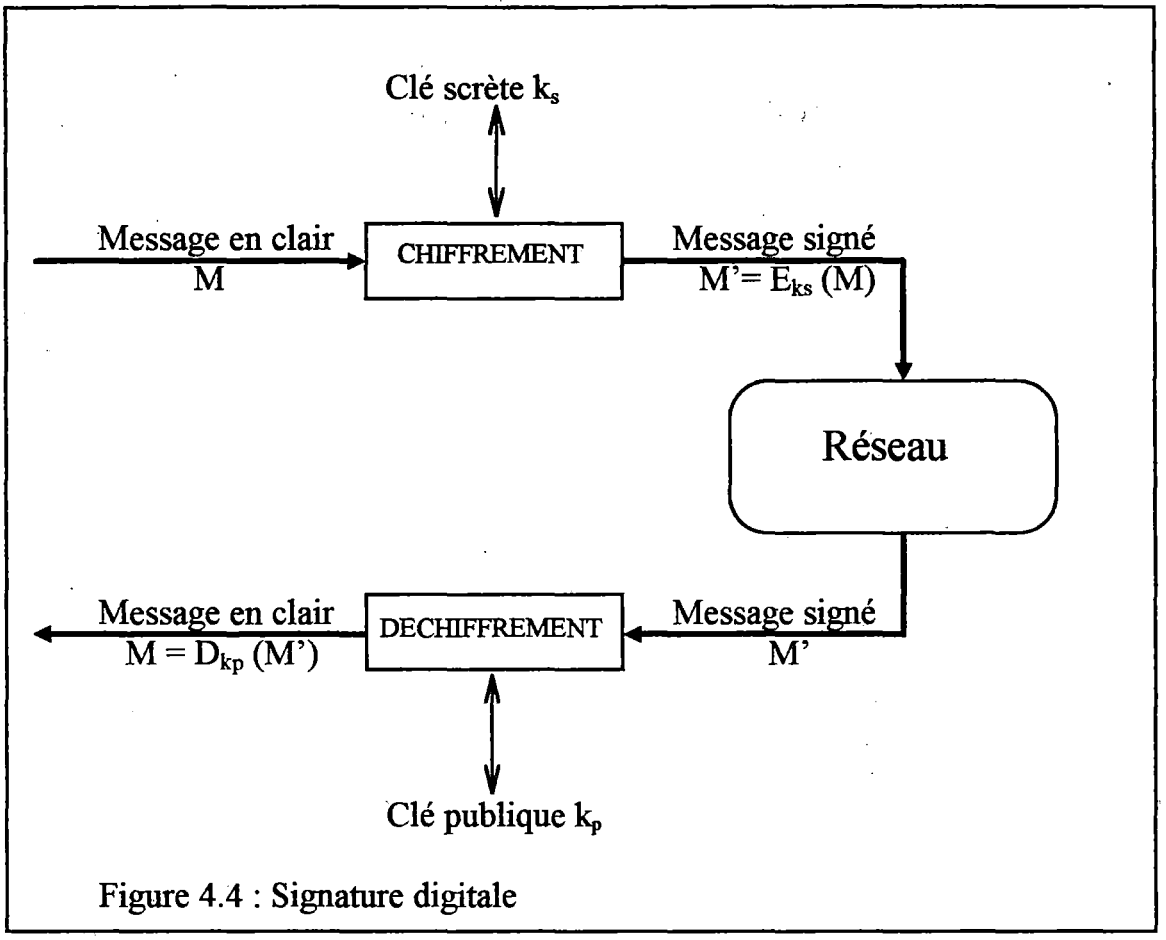

Figure 4.4 : Signature digitale

5 Certificat

5.1 Introduction

Bien qu'il ne faille pas protéger les clés publiques contre leur divulgation aux yeux de tous, il est important de veiller à ce qu'elles ne soient pas falsifiées. Il faut à tout prix avoir la certitude qu'une clé publique appartient bien à la personne qui prétend en être l'initiatrice. Voici un petit scenario qui illustre cet aspect important des méthodes à clé publique et dont on n'avait pas encore parlé :

Un utilisateur A désire envoyer un message à un autre utilisateur B. Le protocole de base du chiffrement à clé publique est le suivant : A doit connaître la clé publique associée à B, soit P_{kB}. A chiffre le message avec la clé publique P_{kB}, ainsi B est le seul à pouvoir déchiffrer le message grâce à S_{kB}. Un « hacker » H peut casser ce protocole de la façon suivante : H génére une clé paire de clé privée/publique : S_{kH} / P_{kH}. Il s'arrange ensuite pour que A utilise P_{kH} au lieu de P_{kB}, en se faisant passer pour B. A ignore tout de la fraude, il chiffre le message avec P_{kH}. Le hacker H est maintenant à même de déchiffrer un message qui ne lui est pas destiné.

5.2 Description technique

Pour empêcher de tels agissements, il faut éviter que quiconque ait l'occasion de falsifier une des clés publiques mises à la dispostion de l'ensemble des utilisateurs : c'est la mission qui est confiée à l'autorité de certification (CA - Certification Authority).

Le CA est un lieu particulier au sein du réseau qui offre des certificats; ceux-ci établissent un lien sûr entre un utilisateur et la clé publique qui lui est associée. De plus, le CA permet à un utilisateur de vérifier l'authenticité de la clé publique qui lui est fournie.

Pour remplir ses fonctions, le CA va signer avec sa clé secrète S_{kCA} le document qui reprend l'identité d'un utilisateur B et sa clé publique P_{kB}. Le certificat qui permettra d'envoyer par la suite un message chiffré à B aura la forme :
$CA(B) = S_{kCA}(id(B), P_{kB})$ avec id(B) qui reprend une pièce d'identité complète de B. Notons qu'optionnellement les certificats peuvent également contenir d'autres informations telles la période de validité de la clé publique ou un identificateur d'algorithme. On pourra vérifier l'authenticité de la clé publique de B grâce à la clé publique de CA, P_{kCA}. Il suffit que chacun des utilisateurs possède au départ une copie sûre de la clé publique de CA.

Bien que l'on ait axé la présentation de cette primitive sur son utilisation dans le chiffrement à clé publique, cette primitive peut également servir à certifier les clés publiques utilisées dans la signature digitale. Pour le cas du chiffrement à clé publique, l'émetteur du message chiffré doit disposer de la clé publique du récepteur souhaité à son message. Pour le cas de la signature digitale, le récepteur du message signé doit disposer de la clé publique de l'émetteur pour pouvoir vérifier la signature sur ce message. Mais dans les deux cas, la clé publique doit être certifiée par le CA.

Jusqu'à présent, nous avons juste décrit le mécanisme à la base de la certification des clés publiques qui est d'ailleurs une simple signature digitale. Les protocoles particuliers, qui permettent à chaque utilisateur de disposer de son propre certificat pour la signature digitale d'un message ou de disposer du certificat du récepteur escompté à son message pour le chiffrement à clé publique, ne seront pas abordés dans ce mémoire, ils font partie du très vaste domaine de la gestion des clés qui justifierait un mémoire à lui-seul. (voir 1.2.4 du chapitre 6).

Chapitre 5

Implémentation des services

Dans ce chapitre, on va décrire un protocole de sécurité pour chaque service de sécurité. Un protocole de sécurité est une sorte de scénario, une démarche à suivre, une façon de parler aux autres utilisateurs qui se servent de primitives de sécurité et qui a pour but de donner une solution à un service de sécurité en particulier. Dans le point 2, on décrira les services que peuvent nous offrir des tierces parties de confiance, les « Trusted Third Party ». Ces services facilitent l'implémentation des services de sécurité.

1 Implémentation des services

Dans ce qui suit, nous décrivons brièvement comment implémenter nos services de sécurité en utilisant les primitives que nous avons décrites au chapitre précédent. En plus des six services dégagés au chapitre 3, on présentera des solutions pour le service qu'on appelle « intégrité de la séquence des messages ». Ce service aborde des problèmes tels la duplication, la suppression ou la perte d'un message mais également le problème du rejeu. Ce dernier demande une petite explication : un émetteur A envoie un message M au récepteur B. Lors de ce processus, un parti tiers fait une copie de ce message M et le retransmet, ceci dans le but de faire croire à B que A a envoyé deux fois le même message, alors qu'il n'en a rien fait. Un ordre de paiement est un exemple manifeste du besoin de protection contre ceci.

1.1 Identification de l'utilisateur

Quand les messages sont sécurisés, il est important d'être capable d'identifier de façon non ambiguë les entités impliquées dans l'échange: l'émetteur qui sécurise le message avant de le transmettre et le récepteur qui réalise les vérifications nécessaires sur le message reçu. Ces entités doivent être identifiées dans les segments de sécurité. Si on utilise des algorithmes asymétriques, cette identification sera réalisée au moyen des certificats, nous commencerons donc par présenter en détail le protocole d'identification dans ce cas précis.

1.1.1 Cas des méthodes à clé publique

Nous allons décrire uniquement en détail comment implémenter la première étape du processus d'identification de l'utilisateur, qui se fait en vérifiant qu'un utilisateur est enregistré avec un certain nombre de pièces d'identité et une clé publique. Comme cela fut déjà mentionné, cette étape est de loin la plus pertinente dans les applications EDIFACT. Elle correspond à la 1-way handshake de X.509, X.509 étant la recommandation pour promouvoir l'authentification des utilisateurs. X.509 décrit également les 2-way handshake (protocole d'authentification d'acteurs par échange de challenge) et 3-way handshake (protocole d'authentification mutuelle

d'acteurs) Beaucoup d'applications EDI ont un délai temporel significatif pour les interactions entre utilisateurs et, par conséquent, nous ne pouvons pas utiliser les 2- and 3-way handshake dans cet environnement. [X.509]

Bien sûr, la façon la plus simple de vérifier si un utilisateur est enregistré est de garder une liste complète des utilisateurs et de leurs pièces d'identité disponible à tout le monde. Cependant, mettre à jour et garder on-line une telle liste est presque toujours ingérable.

Par conséquent, une meilleure solution est celle décrite dans X.509, qui utilise des certificats à clé publique. Ce protocole d'identification de l'utilisateur va donc employer la technique de certification du chapitre précédent. Rappelons que cette solution nécessite une autorité de certification, le CA, qui a obtenu la confiance d'un large groupe d'utilisateurs dès qu'il s'agit de vérifier l'identité de nouveaux utilisateurs. Le CA n'aura pas accès aux clés secrètes des utilisateurs, il fait donc partie de la classe des « functionally Trusted Third Party », qui sera décrite dans le point 2.

Le protocole est le suivant : le CA va signer un message pour chaque utilisateur. Pour l'utilisateur B avec la clé publique P_{kB} et les pièces d'identité (credentials - C_B), le message sera simplement constitué de trois parties :

ID (B), P_{kB}, C_B

où ID (B) est un string identifiant de façon unique un utilisateur (nom, adresse, etc.). Ayant identifié B physiquement, le CA peut alors lui donner ce certificat en le signant par l'intermédiaire de sa clé secrète S_{kCA}. Donc, B reçoit de CA :

S_{kCA} (ID (B), P_{kB}, C_B)

Cela signifie effectivement ceci : « Par le présent document, l'autorité de certification CA certifie que l'utilisateur identifié par ID (B) est enregistré dans le système avec la clé publique P_{kB} et les pièces d'identité C_B ».

Nous assumons que tous les utilisateurs possèdent une copie de la clé publique P_{kCA}. En utilisant celle-ci, tout utilisateur peut vérifier le certificat de tout autre utilisateur.

Un message peut être sécurisé par plusieurs entités, par exemple un message peut avoir des signatures digitales multiples, et par conséquent les informations relatives à la sécurité doivent être répétées pour permettre l'identification de plusieurs entités signataires ou authentificatrices et pour également inclure les différentes signatures digitales et valeurs de contrôles. L'alternative aux méthodes à clé publique est de réaliser certains services de sécurité au moyen des méthodes à clé secrète.

1.1.2 Cas des méthodes à clé secrète

Si les techniques symétriques sont utilisées, alors l'identité des parties impliquées devra être indiquée dans le champ « security sender/recipient name ».

1.2 Intégrité de la séquence des messages

Le service d'intégrité de la séquence des messages protège l'utilisateur contre la duplication, l'addition, la suppression, la perte et le rejeu d'un message.

Pour détecter les messages perdus :
→ L'émetteur doit inclure, et le récepteur vérifier, un « message sequence number ». Ce nombre est relatif au flux de messages entre les deux parties concernées.
→ L'émetteur peut demander un « acknowledgement » et vérifier par après celui-ci.

Pour détecter les messages ajoutés ou dupliqués :
→ L'émetteur peut inclure, et le récepteur vérifier, un « message sequence number ».
→ L'émetteur peut inclure et le récepteur vérifier, un « time stamp ».

Quand on utilise les « sequence numbers », il faut alors se mettre d'accord sur la façon de les gérer.

Le time stamp sera normalement produit par le système de l'émetteur. Ceci implique, comme dans le monde du papier, que la précision initiale de la valeur du time stamp est uniquement sous le contrôle de l'émetteur.

Cependant, il existe des protocoles qui ont recours à un functionally trusted TTP, le TSS (Time-Stamping Service). Décrivons brièvement ce protocole : le client calcule un condensé du message (hash value) et l'envoie au TSS. A partir de ce moment, le TSS ajoute la date et l'heure, signe le document et renvoie le résultat au client. En vérifiant la signature, le client est assuré que le TSS a accédé à la requête, que le condensé a été correctement reçu et que la date et l'heure correctes y sont incluses.

Puisque l'on va intégrer formellement ce time stamp dans le message EDIFACT, il nous importe peu de savoir comment il a été produit, on laissera donc aux utilisateurs l'appréciation du soin à apporter à ce time stamp.

Dans le but d'obtenir une protection totale, l'intégrité du time stamp ou du sequence number doit être garantie par un ou plusieurs des protocoles mentionnés ci-dessous.

1.3 Intégrité

Le service d'intégrité du contenu d'un message protège un utilisateur contre les modifications des données.

L'émetteur réalise cette protection en incluant dans le message une valeur de contrôle d'intégrité. Cette valeur peut être calculée en utilisant un algorithme cryptographique approprié tel le MDC (Manipulation Detection Code). Le protocole est le suivant :

A→B : M et MDC (M)
B : si le MDC recalculé sur M reçu est différent du MDC reçu alors le message M a été manipulé.

Ce n'est pas suffisant, encore faut -il bien protéger la valeur de contrôle MDC (M). Des mesures additionnelles doivent être prises tel l'envoi de la valeur de contrôle par un canal de communication séparé ou le calcul d'une signature digitale sur cette valeur de contrôle. En fait, la signature digitale fournit le service de non-répudiation de l'origine. On peut encore imaginer un troisième protocole basé sur le chiffrement à clé privée :

A→B : E_k (M, MDC (M))

B : M, MDC (M) = D_k (E_k (M, MDC (M)))
si le MDC recalculé sur M reçu est différent du MDC reçu alors le message M a été manipulé.

Une autre alternative à envisager vient du service d'authentification de l'origine qui, s'il est obtenu en utilisant un MAC, impliquera le service d'intégrité du message.

En conclusion, le service d'intégrité du contenu du message dans le cadre de l'EDI sera typiquement obtenu en temps que sous-produit soit du service d'authentification de l'origine du message soit du service de non-répudiation de l'origine.

1.4 Authentification de l'origine

Le service d'authentification de l'origine du message protège le récepteur contre le vrai émetteur de ce message prétendant être une autre entité autorisée.

On peut réaliser une protection contre ces faits en incluant dans le message transmis une valeur authentificatrice, par exemple, une valeur MAC (Message Authentification Code). La valeur dépend du contenu du message et d'une clé secrète en possession de l'émetteur et du récepteur. Le protocole est le suivant :

A→B : M et MAC_k (M)

B : si le MAC_k recalculé sur M reçu est égal au MAC_k reçu alors ce message ne peut provenir que d'un utilisateur en possession de la clé secrète k, soit A. De plus, on est certain que le message n'a pas été altéré par un parti tiers.

Signalons que ce protocole n'offre pas une méthode pour prouver à une tierce partie que le message vient de A, car B lui aussi connaît la clé secrète et il aurait donc pu également produire un tel message (ce cas de fraude s'appelle « forgery »).

Par conséquent, cette solution peut inclure l'intégrité du contenu du message. Ce service sera également obtenu en temps que sous-produit du service de non-répudiation de l'origine.

Dans la plupart des cas, il sera désirable d'avoir au moins l'authentification de l'origine.

1.5 Non-répudiation de l'origine

Le service de non-répudiation de l'origine protège le récepteur d'un message contre la situation où l'émetteur nie avoir envoyé le message.

On peut réaliser une protection contre ces faits en incluant une signature digitale dans le message transmis. Une signature digitale est obtenue par le chiffrement, avec un algorithme asymétrique et une clé secrète, du message ou d'une valeur de contrôle dérivée du message (par exemple, en utilisant une hash function). La signature digitale peut être vérifiée en utilisant la clé publique qui correspond à la clé secrète qui a servi à la créer. Cette clé publique doit être inclue soit dans l'interchange agreement signé par les deux parties, soit dans un certificat signé par une Certification Authority. Le certificat peut être envoyé telle une partie du message lui-même.

Notons que, dans le but de protéger le récepteur contre le problème du rejeu, l'émetteur devra signer non seulement M mais également le time stamp ou le message sequence number. Remarquons que le service d'identification de l'utilisateur et de sa clé publique prend tout son sens dès qu'une méthode à clé publique est employée, en particulier la signature digitale.

La signature digitale ne fournit pas seulement la non-répudiation de l'origine mais également l'intégrité du contenu du message et l'authentification de l'origine.

1.6 Non-répudiation de la réception

Le service de non-répudiation de la réception protège l'émetteur d'un message contre la situation où le récepteur nie avoir reçu le message.

On peut réaliser une protection contre ce fait en demandant au récepteur d'envoyer un « acknowledgement » (un reçu) qui inclut une signature digitale basée sur les données du message originel. La signature digitale porte donc soit sur le message M en entier, soit sur un condensé (« hash value » , MDC (M) par exemple) calculée sur M.

1.7 Confidentialité

Le service de confidentialité du contenu protège un utilisateur contre la lecture, la copie et le dévoilement non-autorisés du contenu du message.

On peut assurer une protection contre ces faits en chiffrant les données. Le chiffrement peut être réalisé en utilisant un algorithme symétrique avec une clé secrète k partagée par l'émetteur et le récepteur.

Cependant, la clé secrète k doit être transmise de manière sécurisée en la chiffrant grâce à la clé publique du récepteur en utilisant un algorithme asymétrique.

Dans un environnement où un système à clé publique a déjà été mis en place pour implémenter le service de non-répudiation de l'origine, pourquoi ne pas chiffrer directement le message avec une méthode asymétrique? La réponse est simple : ce serait tout à fait inefficace. Il suffit de se reporter aux inconvénients de ces systèmes, c'est-à-dire des temps de calculs beaucoup plus important. Une meilleure solution est celle décrite ci-dessus où le système à clé publique, qui est relativement plus coûteux, n'a dû être appliqué qu'à la courte clé secrète k.

1.8 Interrelations entre ces services de sécurité

Comme on a déjà pu s'en rendre compte, certains services embrassent de par leurs solutions d'autres services et il n'est donc pas nécessaire d'inclure dans le message des services qui sont déjà réalisés implicitement. Par exemple, l'utilisation du protocole de non-répudiation de l'origine implique l'intégrité du contenu du message. La table suivante résume ces interrelations :

implique : / l'utilisation de :	Intégrité du contenu	Authentification de l'origine	Non-répudiation de l'origine
Intégrité du contenu	OUI		
Authentification de l'origine	OUI	OUI	
Non-répudiation de l'origine	OUI	OUI	OUI

Figure 5.1 : Interrelations des services de sécurité

2 Trusted Third Party

2.1 Différents types de Trusted Third Party

Quand un groupe d'utilisateurs veut communiquer de façon sécurisée en utilisant des méthodes cryptographiques, il faut prendre des mesures pour distribuer et mettre à jour les clés qui y sont requises. Typiquement, chaque utilisateur doit obtenir une clé provenant d'un autre utilisateur avec qui il veut communiquer, qu'importe le service qui y est requis. Pour un petit groupe toujours composé des mêmes utilisateurs, ceci ne pose pas problème ou très peu, il est d'ailleurs résolu sans impliquer d'autres parties que le groupe d'utilisateurs lui-même.

Pour des groupes d'utilisateurs plus grands ou plus ouverts, le problème devient rapidement difficile et il est alors nécessaire d'impliquer « une tierce partie en qui on a confiance », le Trusted Third Party (TTP).

Malgré l'existence de quelques variantes, il existe une distinction claire que l'on fait habituellement entre deux types de TTPs : **functionally** Trusted Third Parties et **unconditionally** Trusted Third Parties.

Le premier type provient du besoin absolu d'enregistrer convenablement les utilisateurs d'un système. Si les méthodes à clé publique sont utilisées, alors il faut en général procèder à la certification des clés publiques et de leurs appartenances à certains utilisateurs. Un TTP qui a le niveau de confiance pour réaliser cette fonction est appelé « **functionally** trusted ». Il est clair que si l'enregistrement des utilisateurs n'est pas fait d'une manière sûre, alors les utilisateurs ne seront pas sûrs de savoir avec qui ils communiquent. Ainsi, la confiance fonctionnelle représente le montant minimal de confiance qui doit être placé dans le TTP. Notons que ce type de TTP n'a pas besoin ni de connaître la clé secrète d'aucun utilisateur, ni de connaître aucune clé conventionnelle utilisée dans les échanges de données entre utilisateurs. Le CA et le TSS rencontrés plus avant dans ce mémoire sont des exemples de functionally TTP.

Typiquement, on a besoin du second type de TTP uniquement dans les systèmes qui utilisent la cryptographie conventionnelle. En plus de la fonction d'enregistrement mentionné ci-dessus, un « **unconditionally** trusted TTP » va générer des clés pour les échanges de données et ensuite il va les communiquer de façon sûre aux utilisateurs qui en ont besoin. Ceci signifie que le TTP connaît et en principe pourrait faire usage de toutes les informations secrètes voyageant dans le système. Donc, il faut prendre des mesures pour prévenir de tels agissements. Cela implique habituellement l'utilisation de cryptomodule, qui nous assure qu'aucune clé ne va apparaître en clair à l'extérieur de l'environnement sécurisé.

2.2 Services d'un Trusted Third Party

Dans un environnement EDI sécurisé, il faut installer différentes autorités pour remplir des rôles spécifiques plus ou moins directement reliés à l'introduction de la sécurité dans l'EDI. Puisqu'elle dépend d'un environnement, une autorité peut couvrir un ou plusieurs rôles, nous nous focaliserons plutôt sur les différentes tâches que sur les autorités elles-mêmes. Les trois tâches les plus importantes sont :

→ enregistrement d'un utilisateur (registration)
→ certification
→ distribution publique de certificat.

Notons que certains environnements pourraient demander des services additionnels au TTP tels la génération de clé.

2.2.1 Enregistrement sécurisé

Une première exigence pour obtenir un EDI sécurisé est d'enregistrer les utilisateurs de façon sûre. En fait, c'est une base pour permettre une identification correcte de l'utilisateur lors d'un transfert de messages électroniques. Ceci signifie que chaque utilisateur doit donner son identification (par quelque moyens prédéfinis) à une autorité de « registration » et doit y laisser pour être enregistré le nom, l'adresse, les coordonnées additionnelles et, peut-être, les privilèges qui le distinguent.

2.2.2 Certification

La certification d'une clé publique d'un utilisateur est une des tâches les plus importantes, si on veut pouvoir réaliser des signatures digitales.

Lorsque l'on a besoin d'une clé publique d'un utilisateur pour vérifier sa signature, l'on veut être sûr d'être en possession de la vraie clé publique de l'utilisateur. Ceci implique premièrement l'enregistrement sécurisé des clés publiques. Ceci signifie que l'utilisateur est lié d'une façon unique (non ambiguë) à sa clé publique grâce à une autorité de certification (qui doit assurer le lien avec l'autorité d'enregistrement). Ceci est réalisé par les certificats. Ils contiennent les pièces justificatives d'identité (credentials) d'un utilisateur avec sa clé publique correspondance, et, pour garantir l'intégrité du contenu, le tout est signé par l'autorité de certification grâce à sa clé secrète. Il s'en suit qu'il faut posséder la clé publique de l'autorité de certification pour retrouver la clé publique d'un utilisateur particulier. Notons que les certificats peuvent optionnellement également contenir d'autres informations telles la période de validité de la clé publique ou un identificateur d'algorithme.

2.2.3 Distribution publique de certificats

Un service de distribution de certificats rendra accessible toutes les vraies clés publiques à tous les utilisateurs du système. De plus, la maintenance des directories des certificats devra être garantie. Cette maintenance implique :

- ➔ période de validité,
- ➔ listes noires et
- ➔ renouvellement d'une clé par l'intermédiaire d'un nouveau certificat.

PARTIE III :

INTEGRATION

Cette troisième partie a pour but d'analyser l'intégration des services de sécurité dans le standard EDIFACT. Le chapitre 6 propose une analyse au cas par cas, passant en revue tous les services de sécurité ainsi que les différents niveaux du standard EDIFACT où ils pourraient être intégrés. De cette analyse découle un ensemble de schémas d'intégration qui présentent la caractéristique de n'impliquer aucun changement à la syntaxe EDIFACT. D'autre part, les Nations Unies ont publié une série de recommandations pour intégrer la sécurité dans EDIFACT mais uniquement pour le niveau Message et pour certains services de sécurité. Ces recommandations se basent sur deux schémas d'intégration différents : le schéma où la sécurité est intégrée au Message sera présenté au chapitre 7 et le schéma où la sécurité est séparée du Message sera présenté dans le chapitre 8.

Chapitre 6

Analyse de l'intégration

Nous allons procéder en deux étapes : Premièrement, nous allons présenter une analyse globale « historique » de l'intégration de la sécurité dans EDIFACT et deuxièmement, nous allons nous concentrer plus précisément sur l'intégration de la sécurité au niveau du Message (Message level security). En ce qui concerne la sécurité au niveau du Message, on peut encore distinguer deux grandes classes de méthodes complémentaires, celles où la sécurité est intégrée au Message et celles où la sécurité est séparée du Message. Ces deux méthodes feront l'objet chacune d'un chapitre, le septième et le huitième.

1. Analyse

Nous avons qualifié cette analyse de globale mais d'« historique » dans l'introduction de ce chapitre pour préciser, d'un côté, qu'elle s'attaque aux différents niveaux où la sécurité peut être intégrée et qu'elle traite l'ensemble des services de sécurité mais, d'un autre côté, qu'elle a évolué pour donner les solutions actuelles exposées dans le point 2; entre autres, les définitions des segments ont changé mais le fonctionnement global est resté le même.

1.1 Intégration des primitives de sécurité dans EDIFACT

Dans ce qui suit, nous discutons de l'intégration des primitives suffisantes pour implémenter les services de sécurité telle la non-répudiation de l'origine et de la livraison dans le standard EDIFACT. Nous allons également brièvement mentionner l'intégration du service de confidentialité.

Nous sommes premièrement concerné par l'intégration des signatures digitales dans EDIFACT. Ceci signifie qu'un système à clé publique doit être utilisé. Rappelons que de tels systèmes ne sont d'aucune utilité tant que l'authenticité des clés publiques n'est pas assurée. Il faut donc inclure l'utilisation des certificats dans la solution. Dans le point 1.2, nous indiquerons la connexion entre les services de sécurité de la partie II et les solutions techniques décrites ci-dessous.

L'analyse de ce point 1 et la solution résultante reposent sur les principes suivants :

- Signature digitale ou authentification du message dans un Message.
- Signature digitale ou authentification du message dans un Interchange.
- La technique peut être utilisée avec la syntaxe EDIFACT existante.
- Elle permet l'utilisation des algorithmes symétriques et asymétriques.
- Un certificat peut être transmis avec la signature.

- Signatures multiples sur un Message/Interchange.

Avant de décrire l'intégration proprement dite, nous allons donner quelques considérations générales qui serviront de base aux descriptions de ce point 1 :

Algorithmes cryptographiques

L'algorithme cryptographique RSA est utilisé à titre d'exemple pour décrire l'intégration d'une signature digitale dans EDIFACT. D'autres algorithmes que le RSA sont possibles, qu'ils soient asymétriques aussi bien que symétriques, la seule différence est la taille des segments de données qu'ils vont exiger. Le choix de l'algorithme cryptographique dépend du service de sécurité que l'on veut. Si, par exemple, le service de non-répudiation de l'origine n'est pas demandé alors une valeur MAC peut remplacer la signature digitale.

Fonction de filtrage

La sortie des algorithmes cryptographiques doit passer par une fonction filtre pour permettre d'éviter les conflits entre les valeurs binaires résultantes du chiffrement avec l'ensemble des caractères EDIFACT. Par conséquent, on transforme les données de la façon suivante du côté de l'émetteur :

- On représente les données suivant le standard EDIFACT.
- On condense les informations grâce aux hash function et on les chiffre.
- On transforme les résultats du chiffrement suivant un filtre.

Le destinataire doit réaliser ces fonctions dans le sens inverse.

Tailles des segments de données

Les tailles des segments de données dépendent aussi bien des algorithmes cryptographiques que des fonctions de filtrage utilisées. RSA avec un modulo long de 512 bit et une fonction de filtrage qui double le nombre de bit entrant (filtre HEX) sont la base pour fixer la taille des segments de données.

D'autres valeurs sont possibles, si on réalise des ajustements sur les tailles des segments de données.

Exigences légales

Les Messages/Interchanges reçus doivent être gardés par le récepteur dans la forme dans laquelle il les a reçus. Ainsi, il sera capable de répéter la vérification de la signature dans le cas où éclaterait une dispute.

Un accord bilatéral doit être établi entre les différentes parties impliquées dans la communication. Cet accord devra spécifier les conditions pour un transfert effectif. Par exemple, une clause de ce contrat pourrait être que l'émetteur d'un message doit recevoir un reçu (acknowledge) signé par le récepteur avant qu'il puisse être assuré

que le transfert se soit bien déroulé. Une autre possibilité pourrait être que le récepteur n'entreprenne des actions à partir du message reçu uniquement si certains Messages/Interchanges ont une signature digitale incluse.

1.1.1 Eléments inclus dans le processus de signature

Une version améliorée de l'ancien segment AUT et un nouveau segment SIF sont introduits. Rappelons que leur utilisation est requise si la sécurité est désirée au niveau Message ou si la sécurité est désirée et qu'on emploie des mécanismes de transfert de fichier non sécurisé (point 3.3 du chapitre 2). De plus, si on désire utiliser le certificat, un nouveau segment CER le contenant est introduit.

Nous allons commencer par décrire en détail ces trois segments (1.1.1.1). Les conventions pour définir des segments de données ont été énoncées au point 2 du chapitre 2. Nous tenons à faire remarquer que les définitions de segments de données que nous allons présenter ci-dessous ne sont plus d'actualité, il s'agit des définitions de segments provisoires de Nov 90. On présentera des définitions plus récentes de segments, qui datent de février 1994 et qui remplissent les mêmes fonctions, dans les deux chapitres suivants. Cependant, dans un premier temps, ces vieilles définitions vont nous permettre d'expliquer plus facilement et plus clairement le fonctionnement global du processus de signature de Messages ou Interchanges EDIFACT. Pour la sécurité au niveau Interchanges, on a besoin de définir deux nouveaux types de messages INTSIF et INTSGN. Le point 1.1.1.2 examinera l'utilisation des segments et des messages dans le processus de signature.

1.1.1.1 Définitions des segments de données et des nouveaux messages.

a) Définition des segments de données :

AUT Authentification (amélioré)

But : fournir les valeurs authentificatrices.

3801	Certificate identification	M	an..35
C801	Authentification value	M	
4817	Authentification qualifier	M	an..3
4802	Authentification value	M	an..70
4802	Authentification value	M	an..70
4802	Authentification value	M	an..70
4802	Authentification value	M	an..70
4802	Authentification value	M	an..70

3801 Identification du certificat
Le segment AUT inclut une référence au certificat contenant la clé publique applicable de telle façon que la signature puisse être vérifiée même si le segment CER n'est pas inclus dans le Message.

C801 Valeur authentificatrice
Cet élément de donnée composite contient le résultat après utilisation de l'algorithme de chiffrement et la fonction de filtrage. La valeur peut être soit une valeur MAC, soit une signature digitale. L'interprétation en vigueur est donnée par l'élément de donnée simple appelé « Authentification qualifier ».

CER Authentification certificat

But : fournir un certificat.

3801	Certificate identification	M	an..35	A unique identification of the certificat
C817	Party identification	M		
3035	Party qualifier, coded	M	an..3	Certificat « owner »
3039	Party identification, coded	M	an..70	
1131	Code list identifier, coded	C	an..2	
C817	Party identification	M		
3035	Party qualifier, coded	M	an..3	Certificat « issuer »
3039	Party identification, coded	M	an..70	
1131	Code list identifier, coded	C	an2	
C818	Date	M		
2005	Date/time qualifier	M	an..3	Issue date/time
2001	Date, coded	M	n6	
2002	Time	C	n4	
2461	Time zone specifier, coded	C	an..3	
C818	Date	M		
2005	Date/time qualifier	M	an..3	Expiration date/time
2001	Date, coded	M	n6	
2002	Time	C	n4	
2461	Time zone specifier, coded	C	an..3	
C804	Public key	M		Filtered public key
4808	Public key	M	an..70	
4808	Public key	M	an..70	
4808	Public key	M	an..70	
C805	Authentification algorithm	C		Identification of algorithm used by certificat owner
4811	Authentification algorithm, coded	M	an..3	
1131	Code list identifier	C	an2	
C801	Authentification value	M		filtered issuer signature
4817	Authentification qualifier	M	an..3	
4802	Authentification value	M	an..70	
4802	Authentification value	M	an..70	
4802	Authentification value	M	an..70	
4802	Authentification value	M	an..70	
4802	Authentification value	M	an..70	
C805	Authentification algorithm	C		Identification of the issuer algorithm
4811	Authentification algorithm, coded	M	an..3	

1131	Code list identifier	C	an2	
C807	Hash function	C		Identification of the issuer hash function
4815	Hash function, coded	M	an..3	
1131	Code list identifier	C	an2	
C806	Filter function	C		Identification of the issuer filter function
4813	Filter function	M	an..3	
1131	Code list identifier	C	an2	

Le segment est structuré en trois parties :

C817 à C805	qui contiennent les données certifiées.
C801	qui contient la signature du CA (Certification Authority).
C805 à C806	qui contiennent les paramètres du système du CA.

Le certificat proprement dit va de C817 à C801.

3801 Identification du certificat
Une identification unique du certificat réalisée par le CA.

C804 Clé publique
Cet élément de donnée composite contient la clé publique pour les systèmes asymétriques.

C805 Algorithme authentificateur
Cet élément de donnée composite contient le nom de l'algorithme utilisé par le propriétaire de ce certificat lorsqu'il calcule ses valeurs authentificatrices.

C801 Valeur authentificatrice
Cet élément de donnée composite contient les valeurs authentificatrices résultantes. Dans le segment CER, C801 contient habituellement la signature digitale du certificat par le CA. Cette valeur authentificatrice est calculée en utilisant la fonction de hachage et l'algorithme spécifiés en C807 et C805 sur les données allant du tag CER jusqu'au dernier élément de donnée de C805 (le premier des deux) et, ensuite, en utilisant la fonction de filtrage spécifiée en C806.

C805 Algorithme d'authentification
Cet élément de donnée composite contient le nom de l'algorithme utilisé lors du calcul de la signature par le CA.

C807 Fonction de hachage
Cet élément de donnée composite contient le nom de la fonction de hâchage utilisée lors du calcul de la signature digitale par le CA.

C806 Fonction de filtrage

Cet élément de donnée composite contient le nom de la fonction de filtrage utilisée pour passer de la représentation sous forme de caractère en C804 et C801 vers la forme binaire.

SIF System security information

But : fournir des informations sur les systèmes de sécurité employés.

C817	Party identification	M		
3035	Party qualifier, coded	M	an..3	Receiver of Message/Interchange
3039	Party identification, coded	M	an..70	
1131	Code list identifier, coded	C	an..2	
C818	Date	M		Timestamp
2005	Date/time qualifier	M	an..3	
2001	Date, coded	M	n6	
2002	Time	C	n4	
2461	Time zone specifier, coded	C	an..3	
C805	Authentification algorithm	C		Identification of the issuer algorithm
4811	Authentification algorithm, coded	M	an..3	
1131	Code list identifier	C	an2	
C807	Hash function	C		Identification of the used hash function
4815	Hash function, coded	M	an..3	
1131	Code list identifier	C	an2	
C806	Filter function	C		Identification of the used filter
4813	Filter function	M	an..3	
1131	Code list identifier	C	an2	
C809	Receipt request	C		Request for security receipt
4816	Receipt request, coded	C	an..3	
1131	Code list identifier	C	an2	
C808	Security labeling	C		
4817	Security label	M	an..3	
1131	Code list identifier	C	an2	

Le segment SIF fournit au récepteur du message les informations nécessaires à la vérification des valeurs authentificatrices du segment AUT. De plus, ce segment SIF contient un timestamp qui assure l'unicité du message.

C817 Identification de la partie
Cet élément de donnée composite a pour but d'identifier un récepteur.

C818 Date
Cet élément de donnée composite contient le timestamp.

C807 Fonction de hachage

Cet élément de donnée composite identifie la fonction de hachage utilisée dans le processus de signature.

C806 Fonction de filtrage
Cet élément de donnée composite identifie la fonction de filtrage utilisée dans le processus de signature.

C809 Demande d'accusé de réception
Cet élément de donnée composite contient la demande de l'émetteur en vue d'obtenir un accusé de réception sécurisé.

C808 Label de sécurité
Cet élément de donnée composite contient les spécifications de l'émetteur quant au niveau de sécurité.

b) Définition de deux nouveaux types de message :

INTSIF Interchange SIF

La structure d'un message d'information quant à la signature de l'Interchange est :

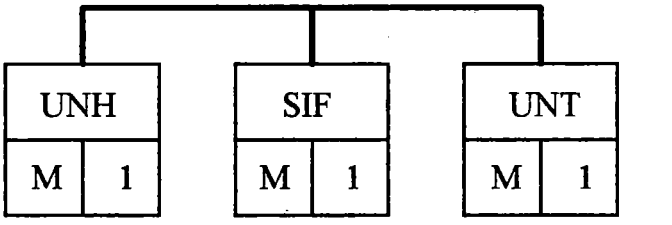

Figure 6.1 : Diagramme de branchement pour le message INTSIF.

Le Message INTSIF indique où débutent les données qui vont être signées quand on signe un Interchange. Le segment SIF fournit un timestamp qui assure l'unicité ainsi que les fonctions de filtrage et de hachage. De plus, une demande d'accusé de réception peut être faite.

Si on utilise des groupes fonctionnels alors le message INTSIF doit être placé dans le groupe fonctionnel.

INTSGN Interchange Signature

La structure d'un message pour la signature de l'Interchange est :

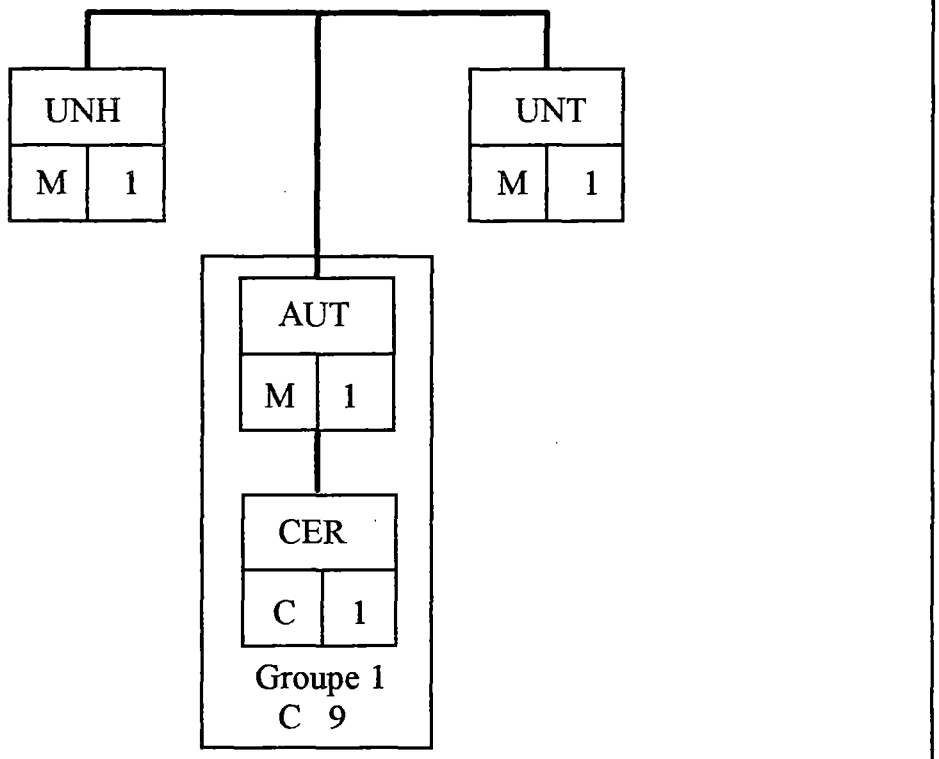

Figure 6.2 : Diagramme de branchement pour le message INTSGN.

Le groupe 1 offre la possibilité de générer des signatures multiples sur un Interchange.

Si on utilise des groupes fonctionnels alors le message INTSGN doit être placé dans le groupe fonctionnel.

1.1.1.2 Utilisation de ces segments de données

a) Signature incluse dans un Message

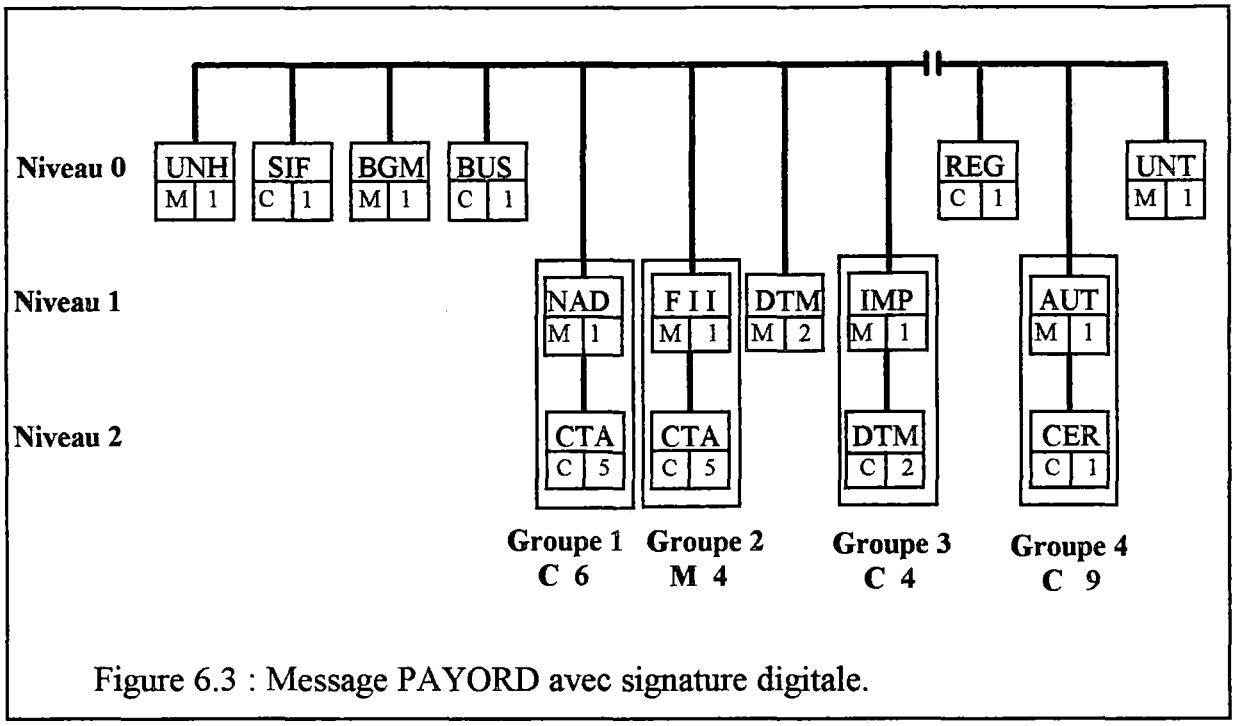

Figure 6.3 : Message PAYORD avec signature digitale.

La signature est calculée sur les données qui vont du segment SIF (inclus) jusqu'au premier segment AUT. Ainsi, les positions respectives du segment SIF et du groupe de segments AUT-CER déterminent quelles sont les données qui sont signées. Par conséquent, les positions les plus favorables sont de faire apparaître le segment SIF le plus tôt possible et le groupe de segments AUT-CER le plus tard possible dans le Message. Par exemple, dans le cas d'une seule signature, le Message entier sera signé si le segment SIF est le premier segment de données du Message et si le groupe AUT-CER est le dernier segment de données du Message. La figure 6.3 montre le diagramme de branchement d'un Message PAYORD qui inclut une signature digitale.

Typiquement, le groupe de segments AUT-CER est utilisé lors de la première transmission à une partie réceptrice et le segment AUT uniquement pour toutes les transmissions futures à cette même partie réceptrice car une référence au certificat est inclus dans le segment AUT. Ce qui est permis par la définition du groupe de segment 4 où le segment CER est facultatif.

Quand on conçoit la structure d'un Message, il faut inclure certains éléments qui assureront que le Message est unique. Ceci est réalisé par l'élément de donnée composite C818 conçu dans ce but et repris dans le segment SIF. Ce timestamp a été introduit spécifiquement pour résoudre le problème du rejeu (replay).

Le segment SIF fournit également une palette d'options pour l'émetteur du Message qui peut demander un accusé de réception, qui peut indiquer le niveau de sécurité requis et qui peut indiquer le récepteur du Message.

Sur la figure 6.3, les segments SIF et le groupe de segments AUT-CER sont facultatifs, ce qui permet d'envoyer des Messages PAYORD non signés.

b) Signature incluse dans un Interchange

Pour signer un Interchange, il faut introduire deux nouveaux types de Message. Un pour indiquer le point de départ (INTSIF) et un autre (INTSGN) pour indiquer l'endroit où se termine la portion signée de l'Interchange. Les valeurs authentificatrices sont placées dans le Message INTSGN. Les données signées vont du Message INTSIF (inclus) jusqu'au dernier Message (inclus) avant le Message authentificateur INTSGN. En fait, si on veut signer un Interchange entier alors le Message INTSIF doit être le premier et le Message INTSGN le dernier. La figure 6.5 présente le schéma pour la signature d'un Interchange.

c) Signature incluse dans un groupe fonctionnel

D'un premier abord, pour les groupes fonctionnels, on pense pouvoir adopter la même approche que pour l'Interchange. La solution demanderait alors un Message INTSIF pour indiquer le début de la zone signée du groupe fonctionnel et un Message INTSGN pour indiquer la fin de cette zone. Cependant, les deux nouveaux Messages qui se chargent de la signature digitale ne sont évidemment pas du même type que ceux du groupe fonctionnel et, par conséquent, ils ne pourront pas être insérés dans le groupe fonctionnel original.

Il n'y a donc pas à proprement parler de signature incluse dans un groupe fonctionnel. Mais si l'on veut apposer une signature à un groupe fonctionnel entier, il faut alors insérer un nouveau groupe fonctionnel dans l'Interchange composé d'un seul Message qui reprend tous les éléments de sécurité pour une signature digitale. Les données signées vont du segment de service UNG du groupe fonctionnel à protéger jusqu'au segment de service UNE de ce même groupe. La figure 6.6 présente le schéma pour la signature d'un groupe fonctionnel.

1.1.2 Evaluation des schémas d'intégration de la signature digitale dans EDIFACT

Sur chacune des figures de cette section, l'en-tête a) présente la structure EDIFACT avant intégration et l'en-tête b) présente la structure qui résulte de l'intégration.

1.1.2.1 Signature digitale dans les Messages EDIFACT

La figure 6.4, qui illustre ce premier schéma d'intégration, présente un Interchange où l'on a signé un seul Message (Msg1). Le nouveau segment AUT amélioré, le segment SIF et un possible segment CER constituent les segments de sécurité de ce schéma.

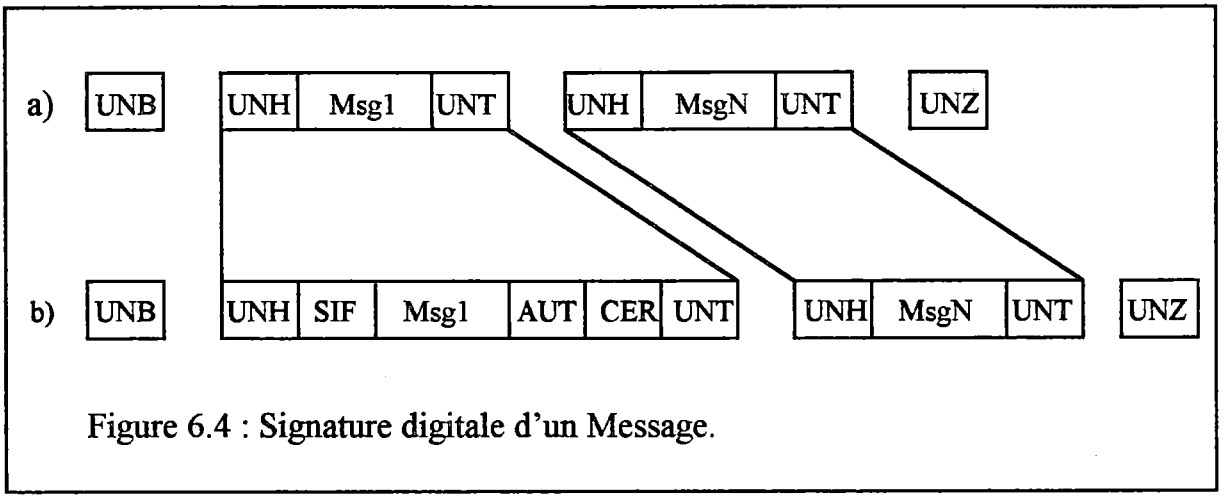

Figure 6.4 : Signature digitale d'un Message.

Les caractéristiques de ce schéma sont :

- Un segment AUT amélioré.
- Un nouveau segment CER pour transmettre son certificat.
- Un nouveau segment SIF pour marquer le début des données signées.

Les avantages de ce schéma sont :

- Les segments pour la signature et pour le certificat peuvent être répétés, si on désire plus d'une signature digitale.
- La signature est naturellement en connexion avec les Messages particuliers auxquels elle appartient. En d'autres mots, les Messages qui ont besoin d'une signature en auront une, les autres pas.
- Aucun conflit avec la syntaxe EDIFACT existante.

Par contre, les désavantages de ce schéma sont :

- Les définitions existantes des Messages doivent être changées pour refléter les nouveaux segments proposés.

1.1.2.2 Signature digitale dans les Interchanges EDIFACT

La figure 6.5, qui illustre ce deuxième schéma d'intégration, considère le cas d'une signature digitale qui couvre tous les Messages d'un Interchange. Deux nouveaux types de Messages, qui contiennent les segments décrits ci-dessus, constituent les éléments de sécurité de ce schéma.

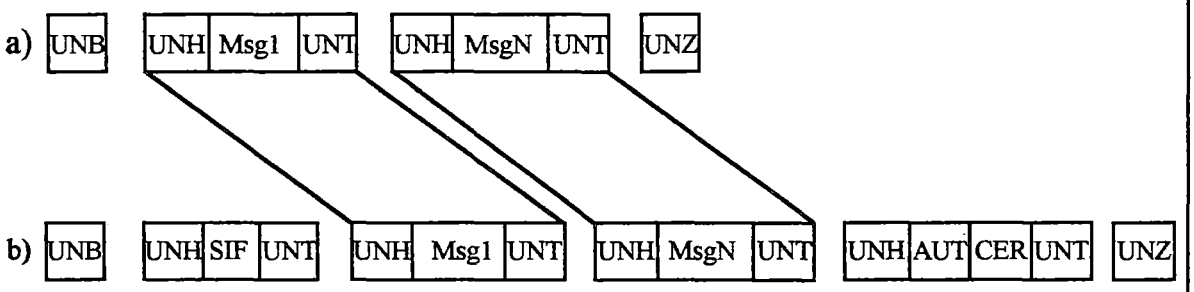

Figure 6.5 : Signature digitale d'un Interchange.

Les caractéristiques de ce schéma sont :

- Deux nouveaux types de Messages (INTSIF et INTSGN).
- Un segment AUT amélioré.
- Un nouveau segment CER pour transmettre son certificat.
- Un nouveau segment SIF pour marquer le début des données signées.

Les avantages de ce schéma d'intégration sont :

- Le Message INTSIGN peut être répété le nombre de fois requis par le nombre de signatures digitales que l'on veut appliquer à l'Interchange (signatures multiples).
- La signature est naturellement en connexion avec l'Interchange auquel elle appartient.
- Aucun conflit avec la syntaxe EDIFACT existante.
- Les utilisateurs qui n'ont pas inclus ces nouveaux éléments dans leurs définitions peuvent encore recevoir et traiter l'Interchange, mais sans pouvoir vérifier aucunement la signature évidemment.

1.1.2.3 Signature digitale dans les groupes fonctionnels EDIFACT.

La figure 6.6, qui illustre ce troisième schéma d'intégration, considère le cas d'une signature digitale qui couvre tous les Messages d'un groupe fonctionnel.

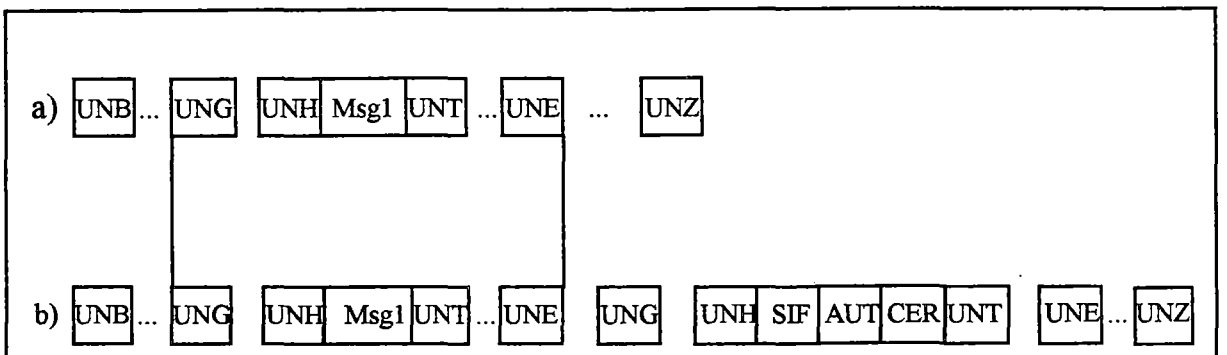

Figure 6.6 : Signature digitale d'un groupe fonctionnel.

De nouveau, ce schéma d'intégration n'est pas en conflit avec la syntaxe EDIFACT existante.

1.1.3 Confidentialité

Puisque la confidentialité n'est pas le service de sécurité le plus demandé pour les Interchanges EDIFACT, nous n'allons que très peu détailler l'intégration d'un tel service dans EDIFACT. Nous présenterons le schéma d'intégration au niveau Interchange et le schéma d'intégration au niveau Message.

1.1.3.1 Confidentialité de l'Interchange

Ce quatrième schéma, qui est illustré à la figure 6.7, décrit un Interchange constitué de plusieurs Messages qui doit être gardé secret vis-à-vis de toutes personnes à l'exception de l'émetteur et du destinataire.

L'Interchange original est enfermé dans un nouveau type de Message qui sera appelé CIPHER. Tous les segments de l'Interchange original, les segments de services aussi bien que les segments de données, seront chiffrés grâce à une clé de session qui sera choisie aléatoirement. Une clé de session est une clé utilisée dans les systèmes à clé secrète et qui est limitée à une seule communication pour des raisons de sécurité. Ces segments chiffrés seront ensuite filtrés et enfermés dans un nouveau Message.

Reste à trouver un moyen sûr pour transmettre la clé secrète (clé de session) à son partenaire commercial. Le protocole suivant va nous dépanner. Nous supposerons ici que l'émetteur ait accès au certificat du récepteur. Avec celui-ci, l'émetteur obtient la clé publique du destinataire avec laquelle on va chiffrer la clé de session. Le résultat est filtré et placé dans un segment de donnée spécial que l'on nomme KEY. Quand le destinataire reçoit l'Interchange, les fonctions inverses doivent être utilisées (filtrage inverse et déchiffrement) pour récupérer en clair la clé de session. Ensuite, le destinataire filtre et déchiffre le segment CPT avec la clé de session. Enfin, le récepteur récupère les segments EDIFACT de l'Interchange original encapsulé dans CPT et il peut traiter normalement cet Interchange. Pour les très longs Interchanges qui vont donner un texte chiffré important, on divisera ce texte chiffré en plusieurs segments de données pour des raisons d'efficacité lors de l'implémentation.

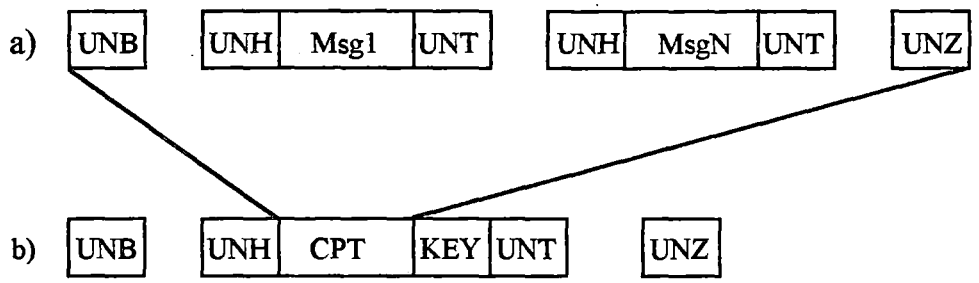

Figure 6.7 : Confidentialité d'un Interchange.

Les caractéristiques de ce schéma d'intégration sont :

- L'en-tête de l'Interchange, qui substitue l'en-tête original lors de la transmission, devra avoir une référence d'Interchange qui est différente de celle de l'original pour éviter les confusions chez le récepteur.
- Un nouveau type de Messages est défini (CIPHER).
- Un nouveau segment de données (qui peut être répété) pour transporter le texte chiffré (CPT).
- Un nouveau segment de données pour transporter la clé de session chiffrée (KEY).
- La référence d'application peut être optionnellement utilisée pour indiquer au récepteur qu'il s'agit d'un Interchange chiffré.

1.1.3.2 Confidentialité du Message et du groupe fonctionnel

Ce cinquième schéma, qui est illustré à la figure 6.8, décrit un Interchange constitué de plusieurs Messages où un seul d'entre eux doit être gardé secret vis-à-vis de l'extérieur. Une approche similaire résoudrait le problème de la confidentialité d'un groupe fonctionnel.

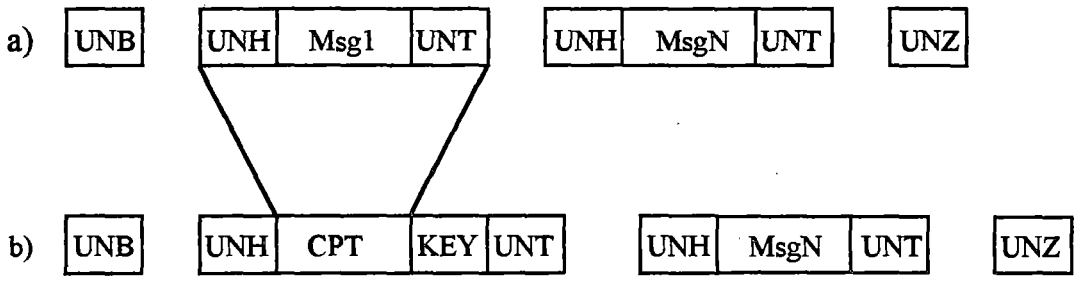

Figure 6.8 : Confidentialité d'un Message.

Les caractéristiques de ce schéma sont :

- Un nouveau type de Messages définis (CIPHER).
- Un nouveau segment de données (qui peut être répété) pour transporter le texte chiffré (CPT).
- Un nouveau segment de données pour transporter la clé de session chiffrée (KEY).

Les avantages de ce schéma sont :

- Il existe des similitudes entre la méthode utilisée pour fournir la confidentialité et la méthode pour signer des Interchanges. Dans les deux cas, des nouveaux Messages ont été créés pour renfermer soit une signature, soit un texte chiffré. Du point de vue implémentation, cela facilitera le développement des fonctions qui seront nécessaires.

1.1.4 ANSI X.12

Dans ce schéma d'intégration, on décrit les concepts d'ANSI mais traduits sous la terminologie EDIFACT.

ANSI X12.58 a introduit quelques segments de service spécifiques à la sécurité dans le but de sécuriser les Messages « EDIFACT ». Les Messages ou groupe de Messages (groupes fonctionnels) à sécuriser débutent et se terminent par des segments de sécurité. Les segments de sécurité de tête (header segments) suivent immédiatement le vrai en-tête du Message (Message header) et le vrai en-tête du groupe fonctionnel. Les « security trailers » sont insérés juste avant les « Message/functional group trailers » d'origine.

La figure 6.9 présente les segments de sécurité au niveau Message uniquement.

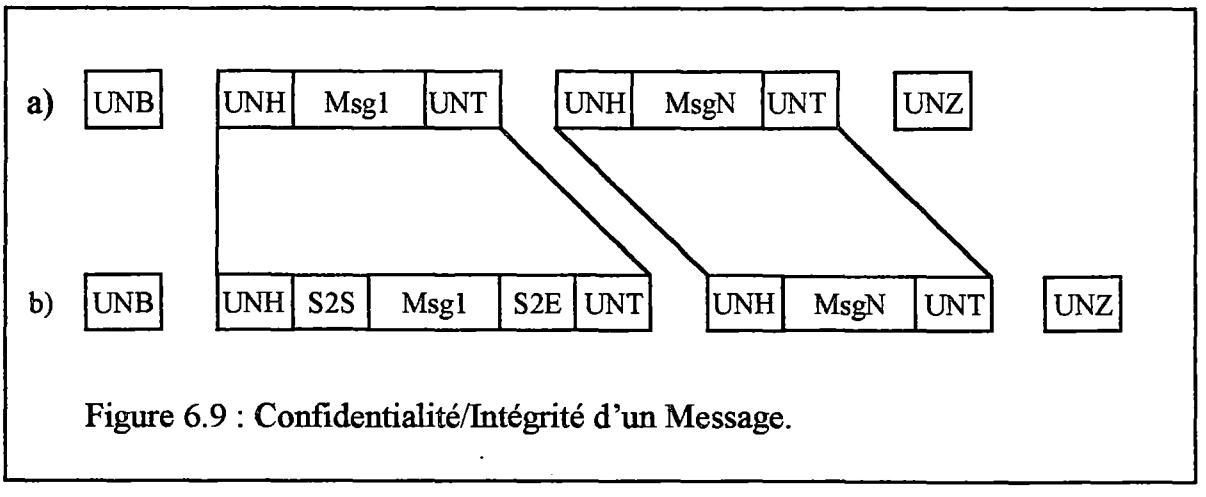

Figure 6.9 : Confidentialité/Intégrité d'un Message.

Les caractéristiques de ce schéma d'intégration sont :

UNB : Interchange Header.
UNZ : Interchange Trailer.
UNG : Functional Group Header.
UNE : Functional Group Trailer.
UNH : Message Header.

UNT : Message Trailer.
SxS : Security Segment Header (suivant le niveau, x prend la valeur 1 ou 2).
SxE : Security Segment Trailer.

Les éléments de donnée suivants sont définis pour le segment SxS :

s01 : Security type (combinaison de l'authentification et de la confidentialité).
s02 : Security Originator.
s03 : Security Recipient.
s04 : Authentification Key Name (nom de la clé utilisée pour l'authentification).
s05 : Authentification Service Code (binary data or coded character string).
s06 : Encryption Key Name (nom de la clé utilisée pour la confidentialité).
s07 : Encryption Service Code (CBC ou CFB-8 et la spécification de la fonction de filtrage).
s08 : Length of Data (llongueur du texte chiffré qui n'est pas filtré).
s09 : Initialization Vector.

L'élément de donnée suivant est défini pour le segment SxE :

e01 : MAC Code.

De plus, certains éléments de données spécifiques sont définis pour les Messages de service en relation avec la gestion des clés.

Si on utilise l'authentification, le segment SxS consiste en les éléments de donnée suivants : Authentification Key Name et Authentification Service Code. De plus, le créateur de la sécurité (security Originator) et le récepteur de la sécurité (Security Recipient) sont spécifiés.

Si on utilise le chiffement, le segment SxS spécifie les éléments de donnée suivants : Encryption Key Name, Encryption Service Code, Initializing Vector et Length of Data

Les avantages de ce schéma d'intégration sont :

- Le standard ANSI possède une structure bien faite et cohérente. Toutes les tailles des éléments de sécurité sont définies. Malheureusement, le standard n'aborde pas le sujet des signatures digitales. Il faudra donc procéder à quelques ajustements de ce standard si l'on veut implémenter des services telle la non-répudiation.
- Les segments de sécurité peuvent également être utilisés au niveau Interchange en plaçant un segment juste après UNB et un autre juste avant UNZ.
- Les services de confidentialité et d'authentification peuvent être choisis indépendamment l'un de l'autre ainsi qu'au niveau désiré : Message, Groupe Fonctionnel ou Interchange).
- Le traitement d'un string tel (SxS (Interchange/Functional Group/Message) SxE) sera le même pour les trois niveaux, facilitant ainsi l'implémentation.

Les désavantages de ce schéma d'intégration sont :

- L'inclusion de ces segments de service spécifiques à la sécurité va impliquer des contraintes sur la syntaxe EDIFACT qui auront des impacts significatifs sur les produits existants.

1.2. Conclusion

Pour conclure cette analyse, nous allons décrire, dans la section 1.2.1, les solutions retenues pour inclure les signatures digitales dans les Messages et Interchanges EDIFACT.

Dans la section 1.2.2, nous reprendrons chaque service de sécurité pour s'assurer que les solutions dégagées sont à même de remplir chacun d'eux.

Ensuite, dans la section 1.2.3, nous fournirons quelques considérations pour choisir au mieux l'algorithme cryptographique et, dans la section 1.2.4, nous donnerons quelques recommandations en ce qui concerne le système d'administration des clés.

Enfin, dans le point 1.2.5, nous porterons un regard critique sur le postulat majeur qui a guidé l'analyse faite au point 1 et qui propose des solutions à l'intégration qui ne demandent aucun changement à la syntaxe EDIFACT.

1.2.1 Description des solutions

Signature digitale dans les Messages EDIFACT

Le segment de sécurité AUT, qui est déjà spécifié dans certains Messages EDIFACT de nature financière, est amélioré pour que la place soit suffisante pour pouvoir y mettre une signature digitale. En plus de la signature digitale, le segment AUT devra inclure un moyen d'identifier le certificat de l'utilisateur qui a généré la signature.

On définit un nouveau segment de donnée SIF qui contiendra un timestamp, et ceci dans le but d'empêcher le rejeu. Il permet également à l'émetteur de remplir un choix d'options telles la demande d'accusé de réception, l'indication du niveau de sécurité exigé et l'indication d'un récepteur.

On définit un nouveau segment de donnée CER qui transportera un certificat. Le segment de certification est défini dans la recommandations X.509 (The Directory - Authentification Framework).

Si on désire apposer plusieurs signatures à un même Message, les segments AUT et CER doivent alors être répétés le nombre de fois nécessaire.

Signature digitale dans les Interchanges EDIFACT

Les segments décrits ci-dessus sont la base pour cette solution également. Cependant, la signature a maintenant pour but de couvrir l'Interchange entier pour éviter de signer chaque Message individuel destiné au même récepteur. Par conséquent, on définit deux nouveaux types de Messages (INTSIF et INTSGN) pour inclure les mêmes segments de données que ceux qui ont été décrits ci-dessus mais la signature s'applique maintenant à tous les Messages dans l'Interchange.

Comme le Message qui contient la signature dépend de tous les messages de l'Interchange, pour des raisons inhérentes à l'implémentation, on insérera le Message INTSGN à la fin de l'Interchange. Si on désire apposer plusieurs signatures sur cet Interchange, alors les nouveaux Messages qui contiennent chaque signature seront ajoutés à la suite les uns des autres à la fin de l'Interchange.

Si le protocole Pedi (défini dans X.435) est utilisé pour le transfert et si l'interface entre l'application et l'agent utilisateur (UA) est sécurisée, alors Pedi peut être utilisé pour fournir la signature digitale.

Les solutions proposées ici pour inclure des signatures digitales dans les Messages et les Interchanges EDIFACT sont très simples à implémenter et n'entrainent aucun conflit avec la syntaxe EDIFACT existante. La seule nouvelle étape pour introduire ces concepts sera de définir de nouveaux types de Messages ainsi que certains segments de donnée.

Un aspect agréable de ces solutions est qu'elles sont suffisamment générales et flexibles pour permettre aux différents groupes d'utilisateurs d'EDIFACT d'implémenter ces concepts dans leurs définitions de Messages.

Signature digitale dans les Messages EDIFACT

On définit un nouveau Message qui contient la signature du groupe fonctionnel. Cette signature est construite de la même façon que pour un Interchange. Ce Message est inclus dans son propre groupe fonctionnel et est placé juste derrière le groupe fonctionnel original qu'il sécurise dans un Interchange.

Confidentialité

Le Message/Interchange chiffré est juste inclus dans un nouveau Message. De plus, la confidentialité et les services de sécurité basés sur la signature peuvent être choisis indépendamment les uns des autres. Le protocole suivant permet la combinaison des deux services : on utilise premièrement le schéma d'intégration de la signature digitale, ce qui produit un Message/Interchange qui inclut une signature digitale. Deuxièmement, ce résultat est utilisé comme entrée dans le schéma pour la confidentialité du Message/Interchange.

1.2.2 Relation avec les services de sécurité

Dans cette section, nous décrivons la connexion entre la solution technique décrite ci-dessus et les services de sécurité mentionnés dans la partie II.

Identification de l'utilisateur

Si on utilise des algorithmes asymétriques alors ce service peut être assuré par l'utilisation des certificats. Un segment de sécurité CER permet d'intégrer le certificat dans le message.

Si on utilise uniquement des algorithmes symétriques, les identités des parties impliquées dans l'échange seront simplement données à l'intérieur de la zone sécurisée du message.

Intégrité de la séquence des messages

Ce service peut être assuré par l'introduction du segment SIF dans le message, en particulier, par les deux éléments de donnée composites suivants : le timestamp et la demande d'accusé de réception.

Intégrité et authentification de l'origine

L'utilisation de la signature digitale par l'émetteur du message remplit les services en question. Cependant, si les services de non-répudiation n'ont pas été demandés pour ce message, on peut alors remplacer la signature digitale incluse dans le segment AUT par une valeur MAC calculée par des algorithmes symétriques.

Non-répudiation de l'origine

Ce service peut être assuré par l'intégration de la signature digitale de l'émetteur dans le message.

Non-répudiation de la livraison

Ce service peut être assuré par le protocole suivant : le récepteur d'un Message doit générer un nouveau Message qu'il va signer et renvoyer à l'émetteur du Message d'origine. Ce nouveau Message contient soit le Message original, soit une valeur de contrôle MDC filtrée générée à partir du Message original. L'émetteur peut, dans le segment SIF, indiquer au récepteur sa demande d'accusé de réception.

Confidentialité

Ce service peut être assuré par les schémas d'intégration de la section 1.1.3.

1.2.3 Choix de l'algorithme cryptographique.

La méthode syntaxique qui est proposée pour intégrer des signatures digitales dans des Messages EDIFACT est, dans une large mesure, indépendante des algorithmes cryptographiques utilisés. La seule restriction est d'ordre logique, il faut que l'algorithme (ou les algorithmes) soit capable d'implémenter les primitives dont nous avons besoin. On ne va pas imposer un ensemble précis d'algorithmes comme étant les seuls qui peuvent être utilisés. Mais l'on va cependant faire une sélection d'algorithmes qui ont les fonctionnalités requises et qui sont considérés comme sûr par un grand nombre d'experts.

Premièrement, si l'on veut implémenter des services telle la non-répudiation, il est clair que cela ne pourra pas se faire sans les systèmes à clé publique. Deuxièmement, vu

que tous les systèmes à clé publique connus sont, de façon significative, plus lents que les meilleurs systèmes conventionnels, beaucoup d'applications seront incapables de fonctionner uniquement sur base de ces systèmes à clé publique, particulièrement dans les situations qui nécessitent de transférer de grands fichiers. Donc, chaque fois qu'une primitive de sécurité n'exige pas d'utiliser un algorithme asymétrique, on utilisera un algorithme symétrique. Par exemple, pour une primitive tel le chiffrement, on privilégiera le chiffrement symétrique. Et troisièmement, pour les primitives qui exigent un algorithme asymétrique, telle la signature digitale, on s'arrange pour réduire la taille des données à sécuriser. Par exemple, on fait appel à une fonction de hachage pour permettre à l'algorithme de signature digitale de n'opérer que sur un condensé d'un message alors que ce dernier peut être très long. Cette fonction peut être construite sur base de cryptosystèmes conventionnels (génération de MDC basée sur le DES), mais il existe aussi des algorithmes dédicacés à cette tâche (MD4 par exemple).

Un choix qui nous vient directement à l'esprit est le RSA pour les systèmes à clé publique et le DES pour les systèmes conventionnels. Ces deux algorithmes sont des standards de facto et offrent la flexibilité appropriée à beaucoup d'applications.

1.2.4 Système d'administration des clés

Pour la sécurité de tous systèmes de traitement de message, une bonne administration des clés est aussi importante que de bons algorithmes cryptographiques. Cependant, les concepts de base des méthodes pour administrer les clés ne sont pas nécessaires pour l'intégration formelle de la sécurité dans EDIFACT , bien qu'ils soient vitaux pour toutes implémentations EDI sécurisées. Nous n'allons donc pas détailler les différents concepts de l'administration des clés qui couvrent les aspects suivants : la génération des clés, la distribution des clés et leurs stockages, l'enregistrement et la certification des clés ainsi que le cycle de vie des clés. Nous allons juste faire une remarque importante.

Lorsque l'on utilise les systèmes à clé publique, il est possible que les utilisateurs ne puissent pas mémoriser leurs clés secrètes, vu que de telles clés sont de très long strings de bits apparemment aléatoires. La solution standard à ce type de problème est de stocker la clé secrète chiffrée par un mot-de-passe que l'utilisateur est capable de se rappeler. Cette solution rencontre bien sûr tous les problèmes classiques des systèmes à mot-de-passe, entre autre le mot-de-passe peut être deviné, volé, ... On améliore grandement le système en stockant la clé dans un object physiquement sûr, comme une carte à microprocesseur (smart card) par exemple. Dans ce cas, un ennemi doit posséder en même temps la carte et le mot-de-passe pour casser le système.

1.2.5 Conclusion

Toutes les solutions proposées dans ce point 1 permettent d'éviter tous changements de la syntaxe EDIFACT. Cependant, on pourrait imaginer une solution cohérente sur base du standard ANSI X.12, si l'on est prêt à accepter de changer la syntaxe EDIFACT. En guise de conclusion et d'introduction au point suivant, nous allons donc retracer l'évolution de la pensée en ce qui concerne la façon d'aborder la sécurité au niveau du Message.

Pour certains types de transfert de Message EDIFACT, on a premièrement constaté que les utilisateurs ont exprimé le besoin de facilité optionnelle en vue d'intégrer des techniques de sécurité.

Ainsi, pendant des années, des groupes de travail tel le Security Joint Working Group (SJWG) ont développé des ébauches de techniques. Cependant, avec les procédures UN/EDIFACT actuelles, l'application de ces techniques au niveau du Message demanderait que, pour chaque Message où la sécurité est demandée, l'on soumette une Data Maintenance Requests (DMRs) pour demander la modification de la structure et du contenu de chaque Message en question. Ce processus est très lent et très perturbateur.

A la suite de discussions entre le groupe pour le développement de la syntaxe (Syntax Development Group - SDG) et le SJWP, il fut reconnu que la résolution à long terme des exigences en matière de sécurité devrait être traitée syntaxiquement. Cependant, comme l'échelle temporelle pour l'approbation de la part de l'UN/ISO de la prochaine révision d'ISO 9735 est très longue, cela ne résout pas les exigences qui nous pressent à court terme.

Par conséquent, les groupes SDG et SJWG furent mandatés pour développer une solution « syntaxique » intérimaire pour la sécurité, qui est disponible en temps que recommandations de l'UN/ECE WP.4 depuis mars 1994.

Simultanément, les groupes SDG et SJWP continuèrent le développement d'une proposition syntaxique qui sera incluse dans la partie 5 de la version 4 de l'ISO 9735.

En plus de tout ceci, ces équipes de travail furent mandatés pour faire du Message AUTACK (anciennement FUNACK) un Message de service avec son propre Service Directory Set. Celui-ci fut soumis à l'UN/ECE WP.4 avec le statut 1 en mars 1994.

Nous allons maintenant présenter brièvement ces deux techniques décrites dans les documents UN/ECE WP.4/R.1026. Le chapitre suivant détaillera la solution syntaxique et le chapitre 8 le Message AUTACK.

2. La sécurité au niveau du Message (Message level security)

Cette section décrit la structure mise en place pour sécuriser EDIFACT au niveau du Message, qui prend en charge la plupart des menaces décrites dans la partie II. Les services de sécurité pourront être soit intégrés dans le Message lui-même ou fournis par un Message séparé.

2.1 La sécurité intégrée au Message.

Tous les services, exception faite de la confidentialité, pourront être fournis par l'inclusion de groupes de segments de sécurité de tête (security header) et de fin (security trailer) après le segment de service UNH et avant le UNT, de telle façon qu'ils pourront être appliqués à un message existant.

Typiquement, il faudra une paire de Message security header/trailer pour chaque service de sécurité demandé.

Les segments de services pour la sécurité (USx), qui seront décrits dans le chapitre suivant, sont essayés en vue d'être inclus dans une version future de la syntaxe d'UN/EDIFACT (ISO 9735). En temps que solution intérimaire et à l'essai, tout utilisateur

qui voudrait incorporer ces segments dans un Message EDIFACT qui se conforme à ISO 9735 (qui date de 1988) peut le faire. Dans ce cas, L'élément de donnée composite « Syntax identifier » (S001) du segment de service UNB (Interchange header) de l'Interchange qui contient ces Messages doit contenir « RTOx » à la place de « UNOx » (x vaut A, B, etc.).

2.1.1 Security Headers et Security Trailers

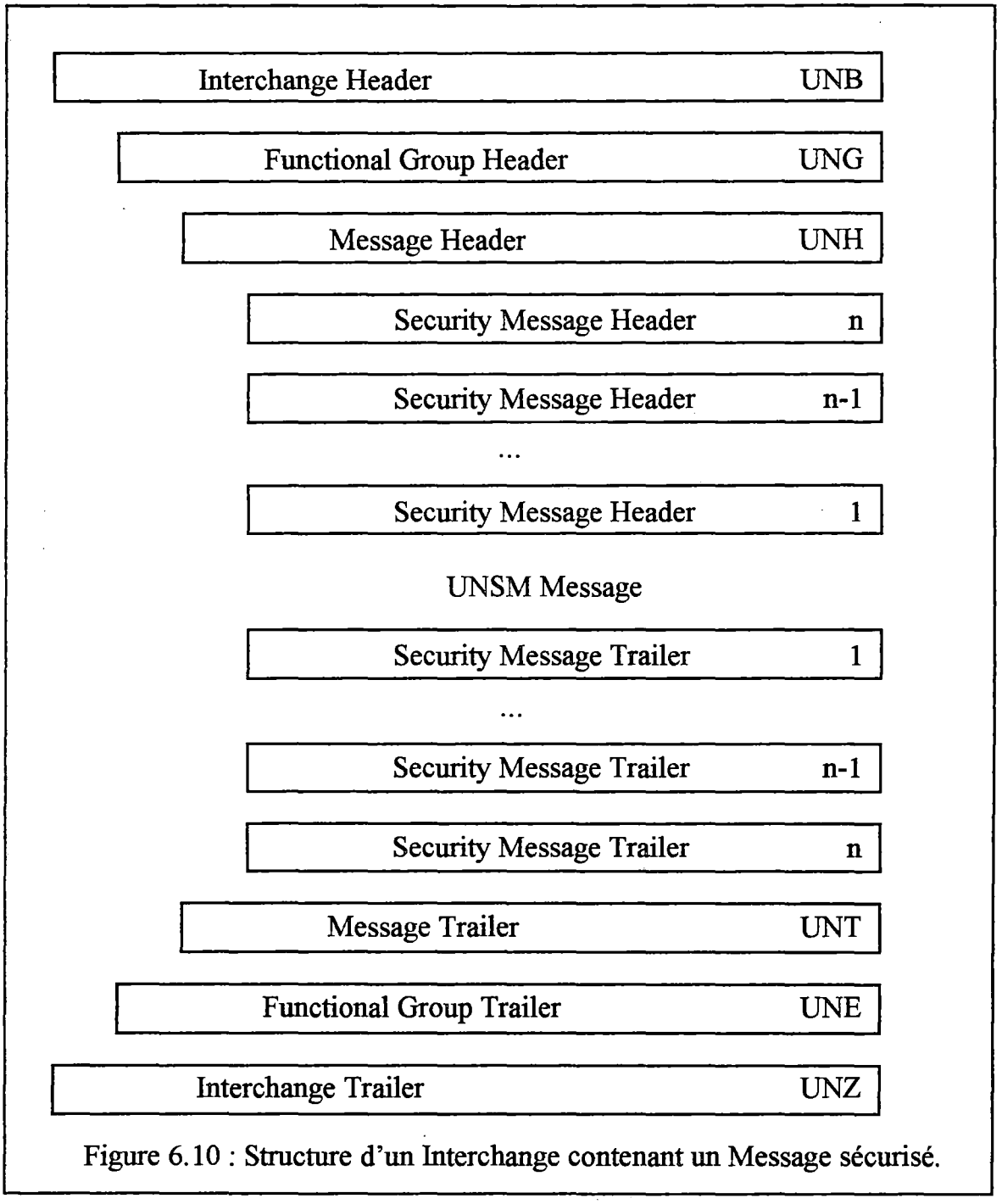

Figure 6.10 : Structure d'un Interchange contenant un Message sécurisé.

Le récepteur d'un Message doit être capable d'identifier et de vérifier les attributs de sécurité associés au Message reçu. Cela implique que les parties impliquées doivent connaître :

- Les services de sécurité impliqués.
- Les mécanismes et les paramètres utilisés.
- La zone sur laquelle s'applique les mécanismes.

Ces informations sont transmises dans des segments spéciaux de sécurité.

La structure d'un Interchange contenant un Message sécurisé est esquissée dans la figure 6.10.

Le but du Message Security Header est de spécifier les méthodes de sécurité appliquées au Message et de garder les données associées qui seront nécessaires pour les calculs de validation. Un segment spécial contient les détails sur les algorithmes de sécurité et d'autres segments contiennent les certificats pour les clés publiques.

Le Message Security Trailer est utilisé pour garder les résultats qui correspondent à l'application des fonctions de sécurité spécifiées dans le Message Security Header associé.

Les Message Security Header et Trailer sont répétés pour chaque couple de service et d'émetteur. Cette approche permet un maximum de flexibilité pour les travaux futurs.

2.1.2 Zones où s'appliquent les services de sécurité

Il existe deux possibilités pour fixer la zone sécurisée :

1. Le calcul de chacune des valeurs authentificatrices et chacune des signatures digitales commence à partir de l'en-tête de sécurité du Message, l'en-tête associée à cette valeur, et inclut le corps du Message lui-même et se termine avec le dernier caractère du Message, juste avant le premier Message Security Trailer.

Par conséquent, l'ordre dans lequel on applique les services de sécurité intégrés de cette manière (sauf la confidentialité) ne doit pas être prescrit. Ils sont complètement indépendant les uns des autres.

2. Le calcul commence à partir de l'en-tête de sécurité du Message (associée aux valeurs calculées) et inclut tous les Security Headers et Trailers ajoutés préalablement ainsi que le corps du Message, dans l'ordre dans lequel ils se présentent.

Pour chacun des services de sécurité que l'on ajoute, on peut choisir l'une ou l'autre approche.

2.2 La sécurité séparée du Message

Deux exigences commerciales appuient le besoin d'une telle possibilité :

1. Pour fournir la sécurité en général, excepté la confidentialité, pour plusieurs Messages dans un seul Message séparé du côté de l'émetteur.

2. Pour fournir un accusé de réception sécurisé à l'émetteur lui indiquant que l'on a bien reçu son Message.

Ces exigences pourront être rencontrées par le « secure authentification and acknowledgement Message » AUTACK.

Les zones sécurisées proposées par AUTACK diffèrent de celles mises en place pour la sécurité intégrée au Message. Elles commencent avant le segment UNH et elles terminent à la fin du segment de service UNT pour les Messages et, de même, avant le segment UNB jusqu'à la fin du segment UNZ pour les Interchanges. En d'autres mots, les en-têtes de sécurité ne sont pas incluses.

2.2.1 L'utilisation par l'émetteur de la sécurité séparée du Message

Cette utilisation d'AUTACK permet à l'émetteur de fournir tous les services de sécurité, à l'exception de la confidentialité, mais envoyés dans un Message séparé. Donc, les services de sécurité peuvent être communiqués plus tardivement ou à un moment plus propice. De plus, il peut prendre en charge plusieurs Messages ou Interchanges originaux, ce qui contraste avec l'intégration directe qui traite un Message ou Interchange à la fois.

Les principes seront identiques pour l'approche séparée que pour l'approche intégrée mais la première exige une référence unique aux Messages originaux qu'ils sécurisent.

2.2.2 L'utilisation par le destinataire de la sécurité séparée du Message

Cette utilisation d'AUTACK rencontre les exigences pour remplir le service de non-répudiation de la réception.

Le Message AUTACK peut être utilisé comme un accusé de réception sécurisé envoyé par le récepteur d'un ou plusieurs Interchanges ou d'un ou plusieurs Messages à l'émetteur de ces derniers. Le Message AUTACK sera construit dans le but de fournir à l'émetteur d'un ou plusieurs Messages ou Interchanges un accusé de réception sécurisé qui prouve que le ou les Messages ont bien été reçus par la partie projetée.

2.3 Principe d'utilisation

2.3.1 Choix du service

Le « Message security header » du Message sécurisé ou du Message AUTACK doit inclure les informations générales suivantes :

- Identification des entités impliquées.
- Identification des mécanismes de sécurité.
- Une valeur « unique » (sequence number ou un timestamp)
- Une demande en vue de la non-répudiation de la réception.

On utilise un code pour indiquer le service de sécurité appliqué.

Si on veut plusieurs services de sécurité à la fois, alors le « Message security header » doit être présent plusieurs fois.

2.3.2 Représentation interne et filtre pour se conformer à la syntaxe EDIFACT.

L'utilisation des algorithmes mathématiques pour calculer des valeurs pour l'intégrité du contenu ou des signatures digitales introduit deux problèmes.

Le premier problème est que le résultat du calcul dépend de la représentation interne des caractères utilisés. Par conséquent, le calcul de la signature digitale par l 'émetteur et sa vérification par le récepteur doivent être exécutés en utilisant la même représentation pour les caractères. L'émetteur doit donc indiquer la représentation utilisée pour produire les résultats.

Le second problème, auquel on a déjà prêté attention, est que le résultat des calculs est un string de bits qui paraît aléatoire. Cela pose problème avec le logiciel d'interprétation. Pour éviter ceci, le string de bits est traduit de façon réversible vers une représentation particulière des caractères au moyen d'une fonction de filtrage. Notons que l'on pourrait éviter d'avoir recours à cette fonction de filtrage dans le futur en adoptant des string de bits où la longueur des données est spécifiée.

2.4 Conclusion

Les deux techniques présentées ci-dessus font l'objet de la recommandation UN/ECE WP.4/R.1026 dont nous allons maintenant résumer les principales caractéristiques.

Cette recommandation pour la sécurité au niveau du Message

- propose des solutions correspondantes à chaque menace quant à la sécurité d'un Message EDIFACT. Les solutions emploient des mécanismes de sécurité qui ont fait leur preuve pour fournir les protections nécessaires.
- permet aux partenaires commerciaux d'implémenter eux-même les services de sécurité, de bout en bout, de façon transparente pour les protocoles de communication sous-jacent, qui peuvent eux aussi fournir des services de sécurité.
- est indépendante des moyens de communication utilisés.
- est un standard ouvert qui supporte l'utilisation de tous les mécanismes de sécurité compatibles avec les services de sécurité identifiés.
- n'implique aucun changement aux Messages individuels. Au contraire, on adopte une approche globale qui peut être appliquée à tous Messages de quelque application commerciale qu'ils soient.

Chapitre 7

La sécurité intégrée au Message

Ce chapitre donne les détails techniques nécessaires pour implémenter la sécurité dans EDIFACT, en accord avec les principes exposés dans le point 2 du chapitre précédent. Ce chapitre est découpé en trois sections. Dans la première section, on spécifie le format adopté pour intégrer la sécurité dans le Message EDIFACT. Ensuite, dans la deuxième section, on spécifie le format et les règles pour chacun des segments de sécurité identifiés. La troisième section nous donne une marche à suivre pour protéger un Message EDIFACT suivant cette technique. Enfin, la quatrième section présente un exemple complet de protection d'un Message.

Ce chapitre présente les segments et les éléments de donnée relatif à la sécurité et inclut les codes et valeurs spécifiques nécessaires pour la sécurité. Cependant, l'ensemble des codes et des valeurs possibles est trop vaste pour être complétement détaillé dans la section 2. Par conséquent, un extrait du document TRADE/WP.4/R.1026/add2, encore appelé « EDIFACT Security Implementation Guidelines », qui liste l'ensemble des codes et valeurs possibles nécessaires à la sécurité défini par le UN/EDIFACT SJWG est présenté en annexe II. Cette annexe II offre donc un complément d'information pour la plupart des éléments de donnée présentés dans la section 2 ainsi que pour ceux repris dans l'exemple de la section 4. On ne fera plus référence à cette annexe II par la suite mais le lecteur pourra toujours s'y plonger utilement dès que le besoin s'en fait sentir. [SIG, 94]

Ce chapitre contient également un exemple de sécurisation de Message. Cet exemple est basé sur le Message PAYORD, qui est un ordre de paiement, décrit dans le manuel MIG des Messages financiers publié par SWIFT. Cependant, les mécanismes de sécurité décrits ici sont totalement indépendants du type de Message et peuvent être appliqués à tout Message EDIFACT.

1. Spécifications du format.

1.1 Liste des segments

SECTION	TAG	DATA SEGMENT NAME	M/C	NO OF REP
SG 1-1		Segment Group 1	C	9
2.1.1	USH	Security Header	M	1
2.2.1	USA	Security Algorithm	C	1
SG 2-1		Segment Group 2	C	2
2.3.1	USC	Certificate	M	1
2.4.1	USA	Security Algorithm	C	3
2.5.1	USR	Security Result	C	1
SG 3-1		Segment Group n	C	9
2.6.1	UST	Security Trailer	M	1
2.7.1	USR	Security Result	C	1

1.2 Diagramme de branchement

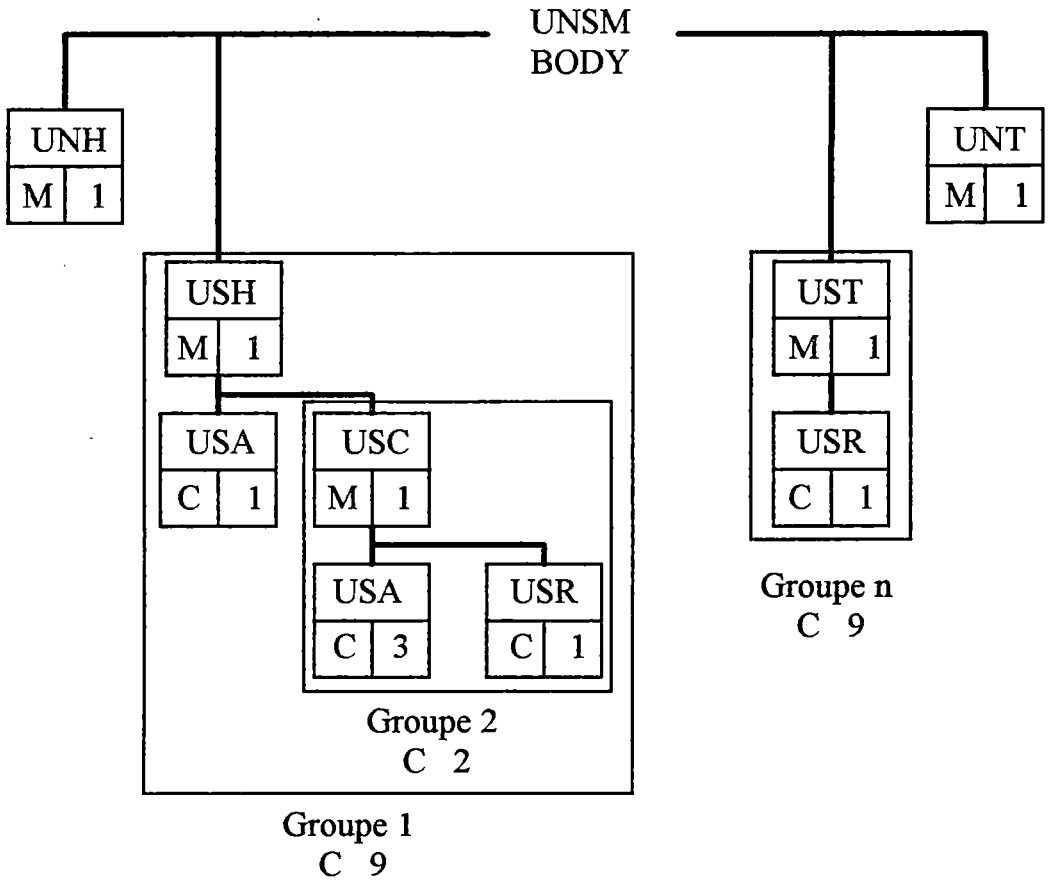

Figure 7.1 : Diagramme de branchement des segments de sécurité

Ce design particulier a été conçu dans l'optique suivante :

- Le groupe de segment 2 est omis à moins que l'on utilise des techniques à clé publique.
- Le segment USA du groupe de segment 1 est relatif aux services de sécurité directement appliqués au Message. Par exemple, pour la signature digitale, ce segment indique la fonction de hachage qui a été utilisée pour ce Message particulier.
- Les systèmes à clé publique exigent toujours au moins un groupe de segment 2, qui est annoncé par le segment USC, même s'il ne contient qu'une référence à une clé publique dans la base de donnée du récepteur. A l'intérieur du groupe de segment 2, les trois segments USA sont pour l'algorithme à clé publique de l'utilisateur, l'algorithme à clé publique de la « Certification Authority » (CA) et pour la fonction de hachage utilisée pour calculer le certificat.
- Le segment UST est utilisé pour séparer le « security trailer » du corps du Message (UNSM).

2. Spécifications des segments

SEGMENT GROUP 1 (Conditional, 9)

USH	Security Header	M	1
USA	Security Algorithm	C	1
	Segment Group 2	C	2
USC	Certificate	M	1

Ce groupe de segment identifie les mécanismes de sécurité appliqués au Message dans lequel ce groupe est inclus. Il inclut l'identification des parties impliquées dans le processus de sécurisation (le créateur des éléments de sécurité et le vérificateur des éléments de sécurité), la référence au Message sécurisé qui inclut la date de création des éléments de sécurité et l'identification des algorithmes de sécurité utilisés. Si l'algorithme utilisé nécessite des certificats, ils seront envoyés dans le groupe de segments 2.

L'algorithme identifié dans le segment USA est l'algorithme directement appliqué au contenu du Message qui est soit un algorithme symétrique (pour les services d'intégrité et d'authentification de l'origine), soit une fonction de hachage (pour la non-répudiation de l'origine).

Si on réalise le service de non-répudiation de l'origine au moyen d'une signature digitale, l'algorithme asymétrique utilisé pour produire la signature devra être identifié dans le segment Certificate si celui-ci est présent ou sera implicitement connu par le récepteur si le certificat n'est pas envoyé dans le Message. Dans ce dernier cas, l'algorithme asymétrique devra être choisi dans l'accord d'échange (Interchange agreement).

2.1 USH - SECURITY HEADER (Mandatory, 1)

2.1.1 Format du segment

Number	Description	M/C	Format	Special notes
0552	SECURITY STRUCTURE VERSION NUMBER	M	an..3	
0501	SECURITY FUNCTION, CODED	M	an..3	
0534	SECURITY RESULT LINK	M	n2	
0541	SCOPE OF SECURITY APPLICATION, CODED	C	an..3	
0503	RESPONSE TYPE, CODED	C	an..3	
0505	FILTER FUNCTION, CODED	C	an..3	
0507	CHARACTER SET ENCODING, CODED	C	an..3	
0509	ROLE OF SECURITY PROVIDER, CODED	C	an..3	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list identifier	C	an..3	

0515	code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list identifier	C	an..3	
0515	code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name .	C	an..35	
0516	SECURITY REFERENCE NUMBER	C	an..35	
S501	SECURITY DATE AND TIME	C		
0517	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	

2.1.2 Description et règles de ce segment

Le segment SECURITY HEADER spécifie le service de sécurité appliqué au Message dans lequel ce segment est inclus. Il peut y avoir plusieurs segments USH différents dans le même Message, si on applique différentes fonctions de sécurité au Message ou si la même fonction de sécurité est appliquée simultanément par plusieurs entités.

L'élément de donnée simple SECURITY STRUCTURE VERSION NUMBER est obligatoire. Il spécifie le numéro de version du format des Security Headers et Security Trailers identifié par l'année et le statut de l'UN/EDIFACT Service Segment Directory.

L'élément de donnée simple SECURITY FUNCTION, CODED est obligatoire. Il spécifie la fonction de sécurité appliquée au Message. Il faut utiliser un des codes suivants :

Code	Mnemo	Signification	Description
1	NRO	Non-répudiation de l'origine	Le Message inclut une signature digitale protégeant le récepteur du refus de la part de l'émetteur d'admettre avoir envoyé ce Message.
2	AUT	Authentification de l'origine	Le véritable émetteur du Message ne peut pas se réclamer être une autre entité autorisée.
3	INT	Intégrité	Le contenu du Message est protégé contre la modification des données.

L'élément de donnée simple SECURITY RESULT LINK est obligatoire. Il contient un nombre qui lie un segment USH particulier à son segment UST correspondant.

L'élément de donnée simple SCOPE OF SECURITY APPLICATION, CODED spécifie la zone sur laquelle on applique le service de sécurité défini dans ce segment USH.

L'élément de donnée simple RESPONSE TYPE, CODED spécifie si on exige de la part du récepteur un accusé de réception sécurisé ou non. Dans le premier cas, l'émetteur du Message s'attend à l'envoi d'un Message AUTACK contenant l'accusé de réception par le récepteur de ce présent Message.

L'élément de donnée simple FILTER FUNCTION, CODED identifie le filtre utilisé pour exprimer les résultats des chiffrements et les clés.

L'élément de donnée simple CHARACTER SET ENCODING, CODED identifie les caractères utilisés pour coder le Message lorsque furent appliqués les mécanismes de sécurité.

L'élément de donnée simple ROLE OF SECURITY PROVIDER, CODED identifie le rôle endossé par le fournisseur de la sécurité. Par défaut, le fournisseur de la sécurité est l'émetteur du document signé.

L'élément de donnée composite SECURITY IDENTIFICATION DETAILS identifie les parties impliquées dans le processus de sécurisation. Deux occurrences de cet élément de donnée sont possibles : une pour l'émetteur de la sécurité et une pour le destinataire de la sécurité.

Si on utilise des algorithmes asymétriques, l'identification des parties est réalisée par l'utilisation des certificats. Cependant, cet élément de donnée sera utilisé :

- si des algorithmes symétriques sont utilisés, ou
- si des algorithmes asymétriques sont utilisés et si deux certificats sont présents, dans le but de distinguer le certificat de l'émetteur et du destinataire.

L'élément de donnée simple *security party qualifier* spécifie la fonction de la partie impliquée :

Code	Mnemo	Signification	Description
1	MS	Message sender	Identifie la partie qui génère les paramètres de sécurité du Message.
2	MR	Message receiver	Identifie la partie qui vérifie les paramètres de sécurité du Message.

L'élément de donnée simple *key name* identifie une clé. En l'absence du segment USA dans le Message sécurisé et en cas d'utilisation d'algorithme symétrique, cet élément de donnée du segment USH permet de nommer la clé privée utilisée.

L'élément de donnée simple *party name* identifie la partie impliquée dans la sécurité.

L'élément de donnée simple SECURITY REFERENCE NUMBER identifie le Message auquel la sécurité est appliquée. Ce nombre de référence peut être utilisé pour fournir le service d'intégrité de la séquence des Messages.

L'élément de donnée composite SECURITY DATE AND TIME spécifie le timestamp qui est appliqué au Message.

2.2 USA - SECURITY ALGORITHM (Conditional, 1)

2.2.1 Format du segment

Number	Description	M/C	Format	Special notes
S502	SECURITY ALGORITHM	M		
0523	Use of algorithm, coded	M	an..3	
0525	Cryptographic mode of operation, coded	C	an..3	
0533	Mode of operation code list identifier	C	an..3	
0527	Algorithm, coded	C	an..3	
0529	Algorithm code list identifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	

2.2.2 Description et règles de ce segment

Le segment est utilisée pour identifier un algorithme, l'usage technique qui en est fait et les paramètres techniques nécessaires à son emploi. Quand on utilise le segment USA du groupe de segment 1, l'algorithme peut être soit symétrique, soit une fonction de hachage. La raison de cette restriction est que les algorithmes asymétriques ne devront pas être référencés directement dans le groupe de segment 1 car ils apparaîtront seulement dans le groupe de segments 2, déclenché par le segment USC.

L'élément de donnée composite SECURITY ALGORITHM est obligatoire. Il se compose des éléments de donnée simples suivants :

L'élément de donnée simple *use of algorithm* spécifie l'usage fait de l'algorithme. Cet élément de donnée doit être utilisé avec les codes suivants :

Code	Mnemo	Signification	Description
1	OHA	Owner hashing	Spécifie que l'algorithme est utilisé par l'émetteur du Message pour calculer le condensé du Message.
2	OSY	Owner symmetric	Spécifie que l'algorithme est utilisé par l'émetteur du Message pour l'intégrité ou l'authentification de l'origine.

L'élément de donnée simple *cryptographic mode of operation* identifie le mode d'utilisation du chiffrement par bloc.

L'élément de donnée simple *algorithm* identifie l'algorithme.

Chaque élément de donnée composite ALGORITHM PARAMETER fournit la place pour un paramètre. Il peut être répété 5 fois. Le nombre de répétitions équivaut au nombre de paramètres nécessaires à l'algorithme utilisé.

L'élément de donnée simple *algorithm parameter value* contient la valeur d'un des paramètres de l'algorithme. Le type précis, l'usage et le format de cette valeur est spécifié dans l'élément de donnée simple *algorithm parameter qualifier* qui le suit.

SEGMENT GROUP 2 (Conditional, 2)

USC	Certificate	M	1
USA	Security Algorithm	C	3
USR	Security Result	C	1

Lorsqu'on utilise des algorithmes asymétriques, ce groupe de segment contient les données nécessaires pour valider les méthodes de sécurité qu'on applique à un Message.

Habituellement, le certificat inclut la clé publique et l'identité de celui qui va posséder ce certificat, ces deux éléments étant signés par une autorité de certification (CA).

On peut soit envoyer entièrement le certificat (le segment USC, les 3 segments USA et le segment USR), soit envoyer uniquement un identificateur de ce certificat (le segment USC identifiera le certificat) qui se réfère à une clé publique qui est déjà connue par les entités impliquées ou qui peut être récupérée dans une base de donnée.

On prévoit deux occurrences de ce groupe de segments, une pour le certificat de l'émetteur du Message (que le récepteur du Message va utiliser pour vérifier la signature de l'émetteur) et l'autre pour le certificat du récepteur du Message (qui ne sera généralement qu'une référence au certificat) dans le cas où la clé publique du récepteur est utilisée par l'émetteur pour rendre confidentiel une clé symétrique.

2.3 USC - CERTIFICATE (Mandatory, 1)

2.3.1 Format du segment

Number	Description	M/C	Format	Special notes
0536	CERTIFICATE REFERENCE	C	an..35	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list identifier	C	an..3	
0515	code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list identifier	C	an..3	
0515	code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0544	FORMAT CERTIFICATE VERSION	C	an..3	
0505	FILTER FUNCTION, CODED	C	an..3	
0507	CHARACTER SET ENCODING, CODED	C	an..3	
0543	CHARACTER SET REPERTOIRE, CODED	C	an..3	
0546	USER AUTHORISATION LEVELS	C	an..35	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S501	SECURITY DATE AND TIME	C		
0517	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	
S501	SECURITY DATE AND TIME	C		
0517	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	

S501	SECURITY DATE AND TIME	C		
0517	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	

2.3.2 Description et règles de ce segment

L'élément de donnée simple CERTIFICATE REFERENCE identifie de façon unique un certificat d'un CA. Ce champ est utilisé pour faire référence à un certificat quand le certificat en entier n'est pas envoyé.

L'élément de donnée simple SECURITY IDENTIFICATION DETAILS identifie les parties impliquées dans le processus de certification. Deux occurences de cet élément de donnée sont possibles : une pour le propriétaire du certificat, une pour le créateur du certificat (CA).

L'élément de donnée simple *security party qualifier* spécifie si cet élément de donnée identifie la partie qui possède le certificat ou s'il identifie la partie qui certifie que le document est authentique.

L'élément de donnée simple *key name* identifie une clé publique : soit la clé publique du propriétaire, soit la clé publique du CA relatives à la clé secrète qui a servi à signer le certificat.

L'élément de donnée simple FORMAT CERTIFICATE VERSION spécifie le numéro de version du certificat identifié par l'année et le statut de l'UN/EDIFACT Service Segment Directory.

L'élément de donnée simple CHARACTER SET REPERTOIRE, CODED identifie le niveau syntaxique utilisé pour créer le certificat, par exemple UNOA, lorsque furent appliqués les mécanismes de sécurité.

L'élément de donnée simple USER AUTHORISATION LEVELS spécifie les privilèges, les niveaux autorisés, etc. associés au possesseur du certificat.

L'élément de donnée composite SEPARATOR FOR SIGNATURE identifie les caractères utilisés comme séparateurs syntaxiques entre les composants du segment USC quand la signature fut calculée. Les séparateurs syntaxiques utilisés dans le Message peuvent être différents de ces caractères. Si cet élément de donnée composite n'est pas présent, les caractères syntaxiques utilisés sont ceux utilisés dans le Message.

L'élément de donnée simple *separator for signature* identifie ces séparateurs. Cependant, pour éviter des problèmes avec le translateur EDIFACT, ces séparateurs passeront dans un filtre hexadécimal. Ainsi, par exmple, le séparateur de segment « ' » sera codé « 27 » si on utilise le codage ASCII 8 bits pour les caractères.

L'élément de donnée simple *separator for signature qualifier* identifie chaque séparateur. Cet élément de donnée prend quatre valeurs distinctes (qui seront codées) : les

séparateurs de segments, d'éléments de données, à l'intérieur d'un élément de donnée composite et pour le caractère d'échappement.

2.4 USA - SECURITY ALGORITHM (Conditional, 3)

2.4.1 Format du segment

Le format de ce segment a déjà été présenté plus avant.

2.4.2 Description et règles de ce segment

Dans le groupe de segment 2, déclenché par le segment USC, trois segments USA sont présents pour identifier trois algorithmes :
1. L'algorithme utilisé par le CA pour calculer le condensé du certificat (hash function).
2. L'algorithme utilisé par le CA pour générer le certificat, c'est-à-dire, pour signer le résultat de la fonction de hachage, elle-même calculée sur le contenu du certificat.
3.a. Soit l'algorithme utilisé par l'émetteur pour signer le Message.
3.b. Soit l'algorithme asymétrique du récepteur utilisé par l'émetteur pour chiffrer la clé secrète requise par l'algorithme symétrique qui sert à sécuriser le contenu du Message.

2.5 USR - SECURITY RESULT (Conditional, 1)

2.5.1 Format du segment

Number	Description	M/C	Format	Special notes
S508	VALIDATION RESULT	M		
0560	Validation value	M	an..256	
0560	Validation value	C	an..256	

2.5.2 Description et règles de ce segment

Le segment USR inclus dans le groupe de segments 2 contient la signature digitale obtenue par le CA en signant le condensé calculé sur les pièces d'identité du bénéficiaire de ce certificat.

L'élément de donnée simple *validation value* contient la signature digitale. Le calcul de la signature commence avec le premier caractère du segment USC (donc « U ») et se termine avec le dernier caractère du dernier segment USA. Cet élément de donnée est filtré par la fonction de filtrage spécifié dans l'élément de donnée FILTER FUNCTION du segment USC. La longueur de cet élément de donnée est déterminée par la longueur de la clé (*algorithm parameter qualifier* modulus length) et par la fonction de filtrage appliquée au résultat du processus de signature.

Dans le cas d'une signature via l'algorithme RSA, un seul élément de donnée *validation value* est nécessaire.

SEGMENT GROUP n (Conditional, 9)

	Segment group n	C	9
UST	Security Trailer	M	1
USR	Security Result	C	1

Ce groupe de segments contient les résultats correspondant aux fonctions de sécurité spécifiées dans le segment USH. A chaque security trailer, on doit faire correspondre un security header dans le groupe de segment 1.

2.6 UST - SECURITY TRAILER (Mandatory, 1)

2.6.1 Format du segment

Number	Description	M/C	Format	Special notes
0534	SECURITY RESULT LINK	M	n2	

2.6.2 Description et règles de ce segment

Ce segment est utilisé pour séparer le corps du Message UNSM du groupe de segment n. L'élément de donnée simple SECURITY RESULT LINK contient un numéro qui relie le segment UST à son segment USH correspondant.

2.7 USR - SECURITY RESULT (Conditional, 1)

2.7.1 Format du segment

Le format de ce segment a déjà été rencontré dans le groupe de segment 2.

2.7.2 Description et règles de ce segment

Les règles ont déjà été décrites et elles restent d'application dans le cadre de ce groupe de segments n.

3. Comment protéger un Message EDIFACT?

La première étape est d'identifier, en coopération avec les partenaires commerciaux, les besoins en matière de sécurité. Les services de sécurité disponibles du côté de l'émetteur sont :

- l'intégrité du contenu du Message,
- l'authentification de l'origine du Message et
- la non-répudiation de l'origine.

Ces services ne sont pas indépendants et il n'est pas nécessaire d'inclure de façon additive les services qui sont réalisés implicitement (la figure 5.1 résume ces inclusions). Par conséquent, l'émetteur devra choisir au plus un des trois services.

La non-répudiation de la réception est un service qui est initié par le récepteur. Il est soit demandé par l'émetteur dans le Message lui-même ou soit mandaté dans un Interchange agreement. Un Message spécial a été développé pour convoyer le reçu.

3.1. Accords bilatéraux ou tierces parties?

Si on veut intégrer des services de sécurité, il faudra bien sûr arranger des accords additionnels avec les partenaires commerciaux. Il existe plusieurs approches possibles dont les deux extrêmes sont brièvement présentés ci-dessous.

Une exigence minimale serait un accord bilatéral avec chacun des partenaires individuels où l'on s'accorde sur les services de sécurité, les algorithmes, les codes, la gestion des clés, les sanctions en cas de mauvaise conduite, etc. Un document provisoire pour faire de tels accords est disponible dans le programme TEDIS de l'EEC. Dans ce cas, on incluera que peu d'informations relatives à la sécurité dans le Message lui-même.

L'autre extrême serait d'impliquer une tierce partie jouant le rôle d'une Certification Authority, qui enregistre tous les utilisateurs et émet des certificats pour les clés publiques des utilisateurs. Dans cette situation, il est plus adéquat d'uniquement conclure un accord avec le CA. Le CA est responsable des listes noires également. Dans ce cas, il faudra inclure de plus vastes informations relatives à la sécurité.

Les services de sécurité ont été intégrés dans le standard EDIFACT de façon à offrir un maximum de flexibilité et à pourvoir aux deux extrêmes décrits ci-dessus ainsi que toutes les situations intermédiaires.

3.2. Aspects pratiques.

Il faut aborder différents aspects pratiques pour pouvoir réaliser ces services de sécurité, tels la génération des clés, la nécessité de disposer d'un traducteur capable de traiter des segments de sécurité, les procédures qui permettront un usage complet des services de sécurité, tel le stockage des Messages signés ou l'utilisation des signatures multiples.

Il est important d'insister sur le fait que l'intégration des services de sécurité est complétement transparent, indépendant des protocoles de communications utilisés. Si un système permet la transmission d'un Message EDIFACT, alors il permettra également la transmission d'un Message EDIFACT sécurisé.

3.3. Procédure pour construire un Message sécurisé.

Premièrement, on crée un Message EDIFACT. Ensuite, on détermine les services de sécurité appropriés à ce Message. Si ces services sont basés sur des signatures digitales, les personnes possédant les clés secrètes seront impliquées. Ceci ne doit pas avoir lieu immédiatement après la génération du Message.

De façon analogue, sur les Messages entrants, la première étape sera de vérifier les services de sécurité et parfois de stocker les Messages sécurisés pour une vérification future.

3.4. Choix des techniques de sécurité.

Une fois les services de sécurité identifiés, les techniques nécessaires pour les fournir doivent être choisies.

Les services de sécurité requis pourront être réalisés soit par une technique, soit par la combinaison de plusieurs techniques. Les techniques les plus importantes ont été décrites dans le chapitre 4. Elles sont également listées et codées pour figurer dans l'élément de donnée *algorithm, coded* des segments USA. De plus, pour chacun des services de sécurité, on a fourni une combinaison des techniques qui garantissait les exigences du service dans le chapitre 5.

La réalisation d'un service de sécurité dépend largement du choix précis de l'algorithme et de ses paramètres. Dans certains cas, un algorithme utilisé avec de mauvais paramètres n'est pas assez sûr, par exemple l'algorithme RSA avec un trop petit modulus. Dans d'autres cas, c'est la combinaison des algorithmes qui amène la fragilisation de l'ensemble du service. Le choix d'un algorithme et de ses paramètres doit être basé sur l'état de l'art actuel des développements cryptographiques.

4. Exemple de protection de Message.

4.1. Message protégé par le service de non-répudiation de l'origine.

Cette exemple va nous montrer comment on peut utiliser les segments de sécurité pour fournir la non-répudiation de l'origine lorsqu'on applique une méthode basée sur un algorithme asymétrique. L'algorithme directement appliqué au Message est un algorithme symétrique, qui requiert un échange de clé secrète entre les partenaires, et qui fournit un MDC. Cette clé symétrique est échangée en l'intégrant dans le groupe de segment d'en-tête, chiffrée au moyen d'un algorithme asymétrique avec la clé publique du récepteur attendu.

Le MDC est signé par un algorithme asymétrique. La clé publique nécessaire au récepteur pour vérifier la signature du Message est incluse dans le premier segment certificate qui est convoyé dans le groupe de segment d'en-tête du Message. Ce certificat est signé par son créateur (le CA appelé AUTHORITY ici) et contient une référence à la clé publique du CA dans le but de permettre à tout partenaire de vérifier l'intégrité et l'authenticité du certificat.

Un second segment certificate contient la référence à la clé publique du récepteur escompté, cette clé est utilisée par l'émetteur pour protéger la clé symétrique.

Cette technique est actuellement utilisée par les banques françaises dans le système ETEBAC 5 (transfert sécurisé de fichier entre les banques et leurs clients).

Cette solution permet à tout partenaire, qui fait confiance au CA et dont il connaît les clés publiques, de vérifier la signature sur le Message reçu en utilisant uniquement des données contenues dans le Message.

4.2. Description du cas

Une compagnie A demande à la banque A, sort code 603000, de débiter son compte de numéro 003877806 du montant de 54345.10 Pounds Sterling le 9 avril 1994. Le montant doit être versé à la banque B, sort code 201827, en faveur du compte de numéro 00663151 de la compagnie B, West Dock, Milford Haven. Le paiement règle la facture 62345. La personne à contacter chez les bénéficiaires est Mr Jones du département des ventes.

La banque veut le service de non-répudiation de l'origine sur l'ordre de paiement de la compagnie A, réalisé par Mr SMITH. La compagnie demande un accusé de réception sécurisé de la part de la banque A (non-répudiation de la réception) qui sera convoyé dans un Message AUTACK.

L'Interchange agreement entre les parties établit que le service de non-répudiation de l'origine devra être fourni pour les ordres de paiements par l'intermédiaire d'une signature digitale. Les deux parties ont pour accord que la signature est calculée par un algorithme asymétrique, RSA de 512 bits, sur un MDC de 64 bits calculé grâce à un algorithme symétrique, DES en mode CBC. Le certificat identifiant la clé publique de Mr SMITH est créé par un CA dans lequel les deux parties ont placé leur confiance.

SECURITY HEADER	
SECURITY STRUCTURE VERSION NUMBER	1994 service segment directory.
SECURITY FUNCTION	Non-répudiation de l'origine.
SECURITY RESULT LINK	La référence de cet en-tête est 1.
RESPONSE TYPE	On demande un accusé de réception.
FILTER FUNCTION	Toutes les valeurs binaires (signatures) seront filtrées au moyen d'un filtre hexadécimal.
CHARACTER SET ENCODING	Le Message était codé en ASCII 8 bits lorsque la signature fut générée.
SECURITY IDENTIFICATION DETAILS Security party qualifier Party name	 L'émetteur du Message. Mr SMITH de la compagnie A.
SECURITY IDENTIFICATION DETAILS Security party qualifier Party name	 Le récepteur du Message. La banque A.
SECURITY REFERENCE NUMBER	Le numéro de séquence pour la sécurité de ce Message est 001.
SECURITY DATE AND TIME	Le timestamp pour la sécurité est : date : 1994 01 15 time : 10:05:30.
SECURITY ALGORITHM	L'algorithme symétrique utilisé pour calculer un MDC.
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation	 L'algorithme de hashing de l'émetteur est utilisé. Cipher Block Chaining : ISO 10116 (n-bits). On calcule un condensé de 64 bits. La valeur d'initialisation est le zéro binaire. Une clé secrète DES est utilisée, elle sera transmise chiffrée par la clé publique de la banque A.

Algorithm	On utilise un algorithme de chiffrement par bloc DES.
ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 La clé symétrique chiffrée par la clé publique de la banque A. Identifie le paramètre d'algorithme précédent comme étant une clé symétrique chiffrée sous une clé publique.
ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 Valeur d'initialisation pour le texte en clair. Identifie le paramètre d'algorithme précédent comme étant la valeur d'initialisation du texte en clair.
CERTIFICATE	Le certificat de Mr SMITH.
CERTIFICATE REFERENCE	Ce certificat est référencé : 00000001 par le CA.
SECURITY IDENTIFICATION DETAILS Security party qualifier Party name	 Le possesseur du certificat. Mr SMITH de la compagnie A.
SECURITY IDENTIFICATION DETAILS Security party qualifier Key name	 Le certificat de Mr SMITH a été généré par le CA que nous appellerons AUTHORITY. La clé publique d'AUTHORITY utilisée pour générer le certificat de Mr SMITH est PK1.
FORMAT CERTIFICATE VERSION	La version du certificat est 94W.
FILTER FUNCTION	Toutes les valeurs binaires (signatures et clés) ont été filtrées par un filtre hexadécimal.
CHARACTER SET ENCODING	Les piéces d'identité étaient codées en ASCII 8 bits lorsque le certificat fut généré.
SEPARATOR CHARACTER FOR SIGNATURE	Quand la signature fut calculée, les séparateurs de caractères étaient : «'» entre les segments, «+» entre les éléments de données, «:» à l'intérieur des éléments de données composites et «?» pour le caractère d'échappement.
SECURITY DATE AND TIME Date and time	La date de génération du certificat. Le certificat de Mr SMITH fut généré le 93:12:15 à 14:12:00.
SECURITY DATE AND TIME Date and time	Le début de la période de validité du certificat. Elle commence en 1994 01 01 000000.
SECURITY DATE AND TIME Date and time	La fin de la période de validité du certificat. Elle se termine en 1994 12 31 235959.
SECURITY ALGORITHM	L'algorithme asymétrique utilisé par Mr SMITH pour signer.
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 L'algorithme de signature de l'émetteur est utilisé. Aucun mode d'utilisation n'est pertinent ici. L'algorithme asymétrique est le RSA.
ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 La clé publique de Mr SMITH. Identifie l'exposant public pour la vérification de la signature.

ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 Le modulo de MR SMITH. Identifie le modulo pour la vérification de la signature.
ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 Le modulo de MR SMITH est long de 512 bits. Identifie la longueur du modulo en bits.
SECURITY ALGORITHM	La fonction de hachage utilisée par le CA pour générer le certificat de Mr SMITH.
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 On utilise l'algorithme de hashing d'AUTHORITY. Square-mod n hash function for RSA CCITT X.509 ISO 9594-8. L'algorithme asymétrique est RSA.
SECURITY ALGORITHM	L'algorithme asymétrique utilisé par le CA pour signer.
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 On utilise l'algorithme de signature d'AUTHORITY. Aucun mode d'opération n'est pertinent ici. L'algorithme asymétrique est RSA.
ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 La clé publique d'AUTHORITY. Identifie l'exposant public pour la vérification de la signature.
ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 Le modulo du CA. Identifie le modulo pour la vérification de la signature.
ALGORITHM PARAMETER Algorithm parameter value Algorithm parameter qualifier	 Le modulo d'AUTHORITY est long de 512 bits. Identifie la longueur du modulo du CA (en bits).
SECURITY RESULT	Signature digitale du certificat.
VALIDATION RESULT	Signature digitale de 512 bits.
CERTIFICATE	Le certificat du récepteur (banque A).
CERTIFICATE REFERENCE	La clé publique de la banque A relative au certificat dont la référence est : 00001001.
SECURITY TRAILER	
SECURITY RESULT LINK	La référence de ce security trailer est 1.
SECURITY RESULT	Signature digitale du Message.
VALIDATION RESULT	Signature digitale de 512 bits.

4.3. Message EDIFACT PAYORD avec le service de sécurité de non-répudiation de l'origine intégré.

No	Segment de donnée	Message
1	Message Header	UNH+326+PAYORD:1:911:RT'
2	Security Header	USH+94W+1+1++2+2+2++1:::::MR.SMITH:COMPANY A+2:::::BANK A+326+1:19940115:100530'
3	Security Algorithm	USA+1:2::1+6E97569DB22F444EA70BAC31A08AD5CDB7C955F2FE0545047F490E2C44FDF33E31E4CAF54EDA66622EF96239A8FCF9B269457FEECC5CC8BBE8AF689B9D29C011:6+0000000000000000:1'
4	Certificate	USC+00000001+3:::::MR.SMITH:COMPANY A+4:PK1:::::AUTHORITY+94W+2+2+++27:1:2B:2:3A:3:3F:4+2:19931215:141200+3:19940101:000000+4:19941231:235959'
5	Security Algorithm	USA+6:0:10+00000003:13+F496F058AB9DCB79491C0A12E5A966259FAAB40202E26793032E34A8D30252F105A78F20DEB376AA1403E302371A8237F737F2A4CDE0227F0E9A62275275BEF7:12+0512:14'
6	Security Algorithm	USA+4:12::10'
7	Security Algorithm	USA+3:0::10+00000003:13+B684FFC6DCC465653E6D2D4E167B8101D28A6266A984D815303E360E204D10C2C3D2F807807500FA2082F364A9F8B4DB79E55571468D945E5EA92C29022392A1E8D:12+0512:14'
8	Security Result	USR+06792DD06115CA63D8A21C05C76D4C1317E6BED21BFBC5119FEFC9FA84FA92B919793A2C8F939F88073E9C53F798CAD0A58FFD8C03208FC7439D75EAB543060E'
9	Certificate	USC+00001001+3:::::BANK A+4:::::AUTHORITY'
10	Beginning of Message	BGM+450+AZ341234+137:920405:101'
11	Name and Address	NAD+BE++COMPANY B:WEST DOCK:MILFORD HAVEN'
12	Contact Informations	CTA++MR.JONES, SALES'
13	Financial Institution Information	FII+OR+00387806:COMPANY A+603000:154:132'
14	Financial Institution Information	FII+BF+00663151+201827:154:101'
15	Date/Time/Period	DTM+203:920414:101'
16	Monetary Amount	MOA+7+9:54345,10:GBP'
17	Document/Message Details	DOC+380+62345'
18	Security Trailer	UST+14
19	Security Result	USR+B531AE86CA202000DC4653B625CA545750352907E7C613DD524A1847A9D4B792BF47FCA768F822D701DD653DCF6ED5AC8CA2C152E45735E82C1910F7B026018CE'
20	Message Trailer	UNT+20+326'

4.4. Quelques explications.

1. La référence de ce Message est 326.
2. La version de la structure de sécurité est celle de 1994 (94W), la fonction de sécurité requise est la non-répudiation de l'origine (code 1), la liaison avec les résultats porte la référence 1, on demande un accusé de réception (code 2), la fonction de filtrage est le filtre hexadécimal (code 2), le codage des caractères est l'ASCII 8 bits (code 2). L'émetteur du Message est Mr SMITH de la COMPANY A et le récepteur escompté est la BANK A. Le numéro de référence pour la sécurité de ce Message est 001, son timestamp (code 1) est : date : 1994 01 15, time : 10:05:30.
3. L'émetteur du Message (Mr SMITH) utilise un MDC (code 1 : OHA) calculé grâce à un algorithme DES (code 1) en mode CBC (code 2) comme condensé du Message. La clé secrète DES utilisée est : 01 23 45 67 89 AB CD EF. Cette clé secrète est chiffrée via une clé publique RSA (code 6 : KYP) dont le modulo est :

   CA 05 6F 9C 89 70 82 00 - E8 22 A8 A1 9B C6 AD D4
   30 80 77 05 DE 2D 5A FA - 79 34 F6 3E A8 E7 C2 80
   37 9C 02 DA 75 87 99 F3 - 4F 2C 0D 1C 27 47 F9 8E
   43 A1 EA A4 BE 81 95 FC - 24 A1 7B E7 04 46 F9 5F

   L'exposant public est fermat4 = 010001 en hexadécimal (65537 en décimal). L'algorithme DES utilise une valeur d'initialisation (code 1 : IVC) qui est nulle. (IV pour les besoins du mode CBC).
4. Ce certificat est référencé par 00000001. Le propriétaire de ce certificat est Mr SMITH de la compagnie A (code 3 : OW). Le certificat a été généré (code 4 : AX) par le CA AUTHORITY qui utilise la clé publique PK1. On a utilisé le format 94W pour le certificat, avec un filtre hexadécimal, le codage ASCII et les séparateurs qui sont explicitement décrit dans le Message. Ce certificat a été émis le 15 décembre 1993 et il est valide entre le premier janvier 1994 et le 31 décembre 1994.
5. Mr SMITH a signé avec un RSA de 512 bits (code 6 : OSG, code 0 : NUL, code 10 : RSA), sa clé publique est 3 (en décimal) et le modulo est : F4 96...BE F7.
6. La fonction de hachage du CA est celle du type square-mod n de l'annexe D du CCITT X.509, ISO 9594-8 (code 4 : IHA, code 12 : SQM, code 10 : RSA). Le modulo utilisé est le même que pour sa signature (voir 7).
7. L'AUTHORITY a signé le certificat avec un RSA de 512 bits (code 3 : ISG, code 0 : NUL, code 10 : RSA), sa clé publique est 3 (en décimal) et le modulo est : B6 84 ... 1E 8D.
8. La signature du certificat.
9. Le certificat de la banque A relatif à la clé publique utilisée pour chiffrer la clé secrète DES est référencé par 00001001. Il a été émis par AUTHORITY.
10. à 17. Ce sont des segments non relatifs à la sécurité.
18. La référence de ce security trailer est 1.
19. La signature de l'ordre de paiement.
20. Le Message de numéro 326 inclut 20 segments.

Chapitre 8

La sécurité séparée du Message

Le Message AUTACK (Secure Authentification and Acknowledgement Message) fournit des services de sécurité sur des Messages envoyés séparément. Ce Message AUTACK peut être utilisé pour toute combinaison de Messages qui ont besoin d'être sécurisée entre deux parties.

Un Message AUTACK, utilisé comme Message d'authentification, est envoyé par l'émetteur des Messages et des Interchanges expédiés séparément ou par une partie qui a autorité pour agir au nom de l'émetteur, pour faciliter l'authentification de l'origine, la validation de l'intégrité du contenu, la validation de l'intégrité de la séquence des Messages ou encore la non-répudiation de l'origine de ces Messages.

Un Message AUTACK, utilisé comme accusé de réception, est envoyé par le récepteur des Messages reçus préalablement ou par une partie qui a autorité pour agir au nom du récepteur, pour faciliter la confirmation de la réception, la validation de l'intégrité du contenu, la validation de la complétude ou encore la non-répudiation de la réception de ces Messages.

Ce Message AUTACK peut s'appliquer à un ou plusieurs Messages ainsi qu'à un ou plusieurs Interchanges.

Les services de sécurité sont fournis par des mécanismes cryptographiques appliqués au contenu du Message ou de l'Interchange original. Les résultats de ces mécanismes forment le corps du Message AUTACK, auxquels on ajoute la référence aux méthodes cryptographiques utilisées, le numéro de référence et la date et l'heure de l'entité originale.

Le Message AUTACK est complété par les groupes de segments standards security header et security trailer.

Dans un premier temps, ce chapitre présente la structure du Message AUTACK et on donne une définition de chaque segment. Ensuite, on précise les principes généraux d'utilisation du Message AUTACK. Finalement, on détaille les segments qui n'ont pas été rencontrés lors de l'étude de la sécurité intégrée au Message, en particulier les codes et valeurs supplémentaires qui s'appliquent tout spécialement au Message AUTACK.

Les spécifications complètes des segments de Message AUTACK peuvent être trouvées dans les documents TRADE/WP.4/R.1026/Add3, « AUTACK : Development of United Standard Messages (UNSMs) », et TRADE/WP.4/R.1026/Add4, « MIG handbook UN/EDIFACT Message AUTACK ». [AUTACK, 94] [MIG, 94]

1. Spécifications du format.

1.1 Table des segments

TAG	NAME	M/C	NO OF REP
UNH	Message header	M	1
----- Segment Group 1 -----		M	99
USH	Security header	M	1
USA	Security algorithm	C	1
----- Segment Group 2 -----		C	2
USC	Certificate	M	1
USA	Security algorithm	C	3
USR	Security result	C	1
USB	Beginning of a security message	M	1
----- Segment Group 3 -----		M	9999
USX	Security references	M	1
USY	Security on references	C	9
----- Segment Group 4 -----		C	9
UST	Security trailer	M	1
USR	Security result	M	1
UNT	Message trailer	M	1

1.2 Diagramme de branchement

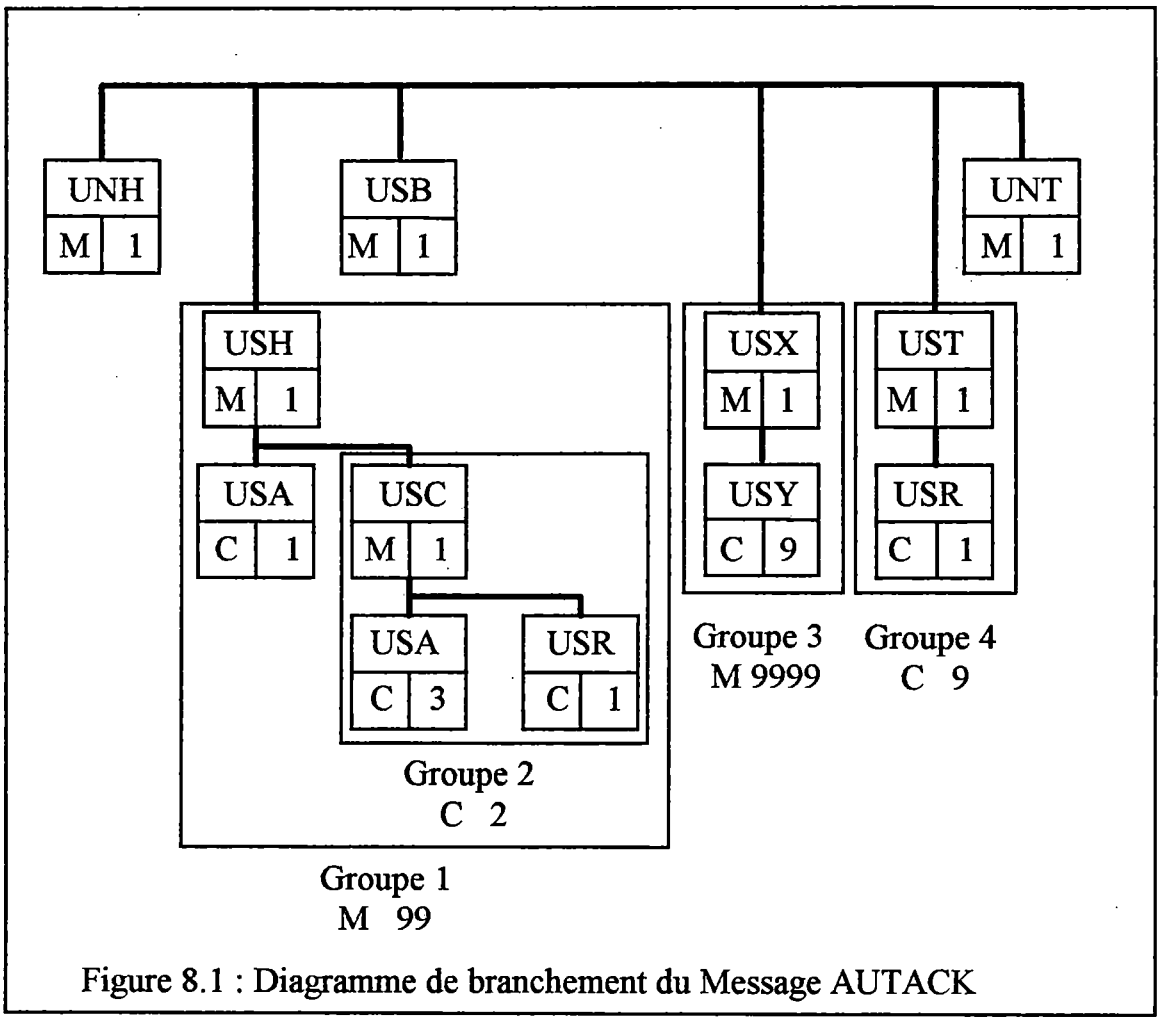

Figure 8.1 : Diagramme de branchement du Message AUTACK

1.3. Définition de chacun des segments.

Le groupe de segments 1 remplit un rôle plus large que dans le chapitre précédent mais chacun de ses segments gardent les mêmes définitions et objectifs, ils ne sont donc pas à nouveau détaillés.

UNH : Message header :
Un segment qui marque le début et identifie de manière unique un Message.

Segment group 1 : USH-USA-SG2 :
Un groupe de segments qui fournit toutes les informations nécessaires pour l'intégrité, l'authentication et la non-répudiation de l'origine ou de la réception de toutes les entités référencées dans le Message AUTACK et du Message AUTACK lui-même.

USB : Beginning of a security Message :
Un segment qui décrit la fonction du Message AUTACK : authentification ou délivrance d'un accusé de réception, la réponse attendue, le timestamp et l'identification de l'émetteur et du récepteur des entités sécurisées.

Segment group 3 : USX-USY :
Un groupe de segments qui identifie les Messages ou Interchanges qui sont sécurisés par le présent Message AUTACK et qui contient les résultats dans l'élément de donnée *validation result.*

USX : Security references
Un segment qui fait référence à l'entité sécurisée et à sa date et heure de création.

USY : Security on references
Un segment utilisé pour repérer l'en-tête qui s'applique (voir point 2), pour contenir les résultats des calculs de sécurité et, dans certains cas, pour indiquer une cause de rejet des valeurs de sécurité référencées.

Segment group 4 : UST-USR :
Un groupe de segments contenant les résultats des méthodes de sécurité appliquées au présent Message AUTACK.

UNT : Message trailer :
Un segment de service qui marque la fin d'un Message et qui donne le nombre total de segments dans le Message et le numéro de référence de ce Message.

2. Principes généraux

2.1. Le Message AUTACK pour l'intégrité, l'authentification et la non-répudiation de l'origine.

2.1.1. Le service d'intégrité

L'entité sécurisée (Message ou Interchange) est référencée dans une occurrence du segment USX. Au dessous de ce segment, il faut au minimum une occurrence du segment USY se référant à un groupe d'en-tête du présent Message AUTACK par l'intermédiaire de l'élément de donnée *security result link.* Cet en-tête particulier contient uniquement la description d'une méthode de hashing, qui est celle qui est appliquée à l'entité référencée depuis son premier segment de service jusqu'à son dernier. Dans cet en-tête, l'élément de donnée *security function, coded* aura le code RINT (Referenced entity Integrity). Dans ce cas, le Message AUTACK doit être sécurisé (signé) par l'intermédiaire d'au moins un nouveau header et de son trailer correspondant. Le point 5.1.3 du chapitre 5 donne la raison qui nous a poussé à signer le Message AUTACK. Il est conseillé d'envelopper l'en-tête utilisé pour référencer la méthode de hashing dans l'en-tête qui sert à sécuriser le Message AUTACK.

2.1.2. Les services d'authentification et de non-répudiation de l'origine

L'entité sécurisée (Message ou Interchange) est référencée dans une occurrence du segment USX. Au dessous de ce segment, il faut au minimum une occurrence du segment

USY se référant à un groupe d'en-tête du présent Message AUTACK. Cet en-tête particulier contient la méthode complète pour ces deux services de sécurité (signature), cette méthode est appliquée à l'entité référencée depuis son premier segment de service jusqu'à son dernier. Dans cet en-tête, l'élément de donnée *security function, coded* aura le code RAU ou RNO (Referenced entity Authentication ou Referenced entity Non-repudiation of origin). Dans ce cas, il n'est pas nécessaire de sécuriser le Message AUTACK.

2.2. Le Message AUTACK pour la reconnaissance ou le refus de la réception.

Un même Message AUTACK peut uniquement accepté ou refusé la réception. Dans les deux cas, le Message AUTACK doit être sécurisé. Au dessous du segment USX déterminant l'entité traitée, il faut au moins une occurrence du segment USY. Cette fois-ci, l'élément de donnée *security result link* s'applique à référencer la méthode de sécurité utilisée dans le Message original (INT, AUT ou NRO), sous le même numéro. L'élément de donnée *validation result* du segment USY va reprendre les valeurs trouvées dans l'élément de donnée *validation result* portant le même numéro dans le Message original.

Dans le cas du refus d'acceptation, un code d'erreur sera fourni dans l'élément de donnée *security error, coded.* Par défaut, la valeur de ce code est 0 signifiant « pas d'erreur du point de vue sécurité ».

Il faut apliquer au moins une méthode de sécurité au Message AUTACK pour authentifier l'acceptation ou le refus d'acceptation de la réception.

3. Spécifications des segments

3.1 UNH - MESSAGE HEADER (Mandatory, 1)

3.1.1 Format du segment

Number	Description	M/C	Format	Special notes
0062	MESSAGE REFERENCE NUMBER	M	an..14	
S009	MESSAGE IDENTIFIER	M		
0065	Message type identifier	M	an..6	« AUTACK »
0052	Message type version number	M	an..3	« 1 »
0054	Message type release number	M	an..3	« 1 »
0051	Controlling agency	M	an..2	« UN »
0057	Association assigned code	C	an..6	
0068	COMMON ACCESS REFERENCE	M	an..35	
S010	STATUS OF THE TRANSFER	C		
0070	Sequence message transfer number	M	n..2	
0073	First/last sequence message transfer indication	C	a1	

3.1.2 Description et règles du segment

L'élément de donnée simple MESSAGE REFERENCE NUMBER identifie un Message par un numéro unique. Le premier Message est 1, le second 2, etc.

L'élément de donnée composite MESSAGE IDENTIFIER spécifie le type de Message envoyé (PAYORD, par exemple), le numéro de version, etc.

L'élément de donnée simple COMMON ACCESS REFERENCE sert de clé pour référencer tous les tranferts de données du même fichier commercial.

L'élément de donnée composite STATUS OF TRANSFER spécifie que le Message est un Message d'un séquence de transfert relatif au même thème.

SEGMENT GROUP 1 (Mandatory, 99)

USH	Security Header	M	1
USA	Security Algorithm	C	1
	Segment Group 2	C	2
USC	Certificate	M	1

3.2. USH - SECURITY HEADER (Mandatory, 1)

Ce segment est inchangé mis-à-part que l'élément de donnée *security function, coded* offre un choix de code plus large :

Code	Mnemo	Signification	Description
1	NRO	Non-répudiation de l'origine	Le Message inclut une signature digitale protégeant le récepteur du refus de la part de l'émetteur d'admettre avoir envoyé ce Message.
2	AUT	Authentification de l'origine	Le véritable émetteur du Message ne peut pas se réclamer être une autre entité authorisée.
3	INT	Intégrité	Le contenu du Message est protégé contre la modification des données.
5	NRR	Non-répudiation de la réception	Le récepteur ne peut pas nier avoir reçu un certain Message.
6	AUR	Authentification de la réception	Le récepteur authentifie qu'il a reçu un certain Message.
7	RNO	RNO sur une entité référencée	Service qui assure la non-répudiation de l'origine sur uneentité référencée.
8	RAU	AUT sur une entité référencée	Service qui assure l'authentification d'une entitée référencée.
9	RINT	INT sur une entité référencée	Service qui assure l'intégrité du contenu d'une entité référencée.

Les fonctions de sécurité NRO, AUT et INT sont appliquées sur le Message AUTACK lui-même. Tous les autres fonctions de sécurité sont connectées aux entités

référencées. NRR et AUR sont des fonctions de sécurité du Message AUTACK utilisées par le récepteur et RNO, RAU et RINT par l'émetteur.

Par conséquent, l'élément de donnée *security result link* va contenir un nombre qui lie un segment USH particulier à son USR ou USY correspondant et admet maintenant les règles suivantes : Si on utilise NRO ou AUT, le lien est USR (groupe 4), si on utilise NRR, AUR, RNO, RAU ou RINT, le lien est USY (groupe 3).

L'élément de donnée *scope of security application, coded* admet le code 3 (TOT) en supplément des deux codes déjà rencontrés (code 1 : SHM et code 2 : SHT). Ce code 3 définit une zone de sécurité qui s'étend sur l'ensemble du Message ou de l'Interchange.

3.3. USB - BEGINNING OF A SECURE MESSAGE (Mandatory, 1)

3.3.1 Format du segment

Number	Description	M/C	Format	Special notes
0563	MESSAGE FUNCTION, CODED	C	an..3	
0503	RESPONSE TYPE, CODED	C	an..3	
S501	SECURITY DATE AND TIME	C		
0517	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	
0590	RELATED FILENAME	C	an..256	
S002	INTERCHANGE SENDER	C		
0004	Sender identification	C	an..35	
0007	Partner identification code qualifier	M	an..35	
0008	Address for reverse routing	C	an..14	
S003	INTERCHANGE RECIPIENT	C		
0010	Recipient identification	C	an..35	
0007	Partenr identification code qualifier	C	an..14	
0014	Routing address	C	an..14	

3.3.2 Description et règles du segment

L'élément de donnée MESSAGE FUNCTION, CODED utilise un des codes suivants :

Code	Mnemo	Signification	Description
1	AUT	Authentification	Le Message est utilisé pour l'authentification et/ou la non-répudiation de l'origine.
2	ACK	Acknowledgement	Le Message est utilisé pour marquer l'acceptation d'un Message reçu.
3	NA	Non-	Le Message est utilisé pour marquer le refus et pour

		Acknowledgement	mentionner les erreurs de sécurité.

L'élément de donnée simple RESPONSE TYPE, CODED est inchangé.

L'élément de donnée composite SECURITY DATE AND TIME identifie le timestamp du Message sur lequel on applique la sécurité, le code supplémentaire suivant peut être utilisé :

Code	Mnemo	Signification	Description
5	MGT	Message generation d/t	La date et l'heure auxquelles l'entité sécurisée fut générée.

L'élément de donnée simple RELATED FILENAME contient le nom du fichier pour lequel on place une signature digitale dans le groupe de segments 3.

L'élément de donnée composite INTERCHANGE SENDER identifie l'émetteur de l'entité sécurisée. L'élément de donnée composite INTERCHANGE RECIPIENT identifie le récepteur de l'entité sécurisée.

SEGMENT GROUP 3 (Mandatory, 9999)

USX	Security references	M	1
USY	Security on references	C	9

3.4. USX - SECURITY REFERENCES (Mandatory, 1)

3.4.1 Format du segment

Number	Description	M/C	Format	Special notes
0020	INTERCHANGE CONTROL REFERENCE	M	an..14	
0062	MESSAGE REFERENCE NUMBER	C	an..14	
S501	SECURITY DATE AND TIME	C		
0517	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	

3.4.2 Description et règles du segment

L'élément de donnée simple INTERCHANGE CONTROL REFERENCE contient la référence unique assignée à un Interchange par l'émetteur.

L'élément de donnée simple MESSAGE REFERENCE NUMBER contient le numéro unique de référence assigné à un Message par l'émetteur.

3.5. USY - SECURITY ON REFERENCES (Conditional, 9)

3.5.1 Format du segment

Number	Description	M/C	Format	Special notes
0534	SECURITY RESULT LINK	M	n2	
S508	VALIDATION RESULT	C		
0560	Validation value	M	an..256	
0560	Validation value	C	an..256	
0571	SECURITY ERROR, CODED	C	an..3	

3.5.2 Description et règles du segment

L'élément de donnée simple SECURITY RESULT LINK contient un numéro qui lie les résultats au segment USH correspondant pour les services RNO, RAU et RINT alors que pour les services NRR et AUR, ce numéro est celui qui identifie la signature sur le Message référencée.

L'élément de donnée composite VALIDATION RESULT contient les résultats correspondant à la fonction de sécurité spécifiée dans le segment USH lié.

L'élément de donnée simple SECURITY ERROR, CODED utilise un des codes suivants :

Code	Mnemo	Signification	Description
0	NULL	No error	
1	NAUT	Wrong authentificator	La valeur de validation est fausse.
2	NCER	Wrong certificate	Le certificat est faux.
3	NPATH	Certificate path	Le chemin d'accès du certificat est incomplet. Impossible de vérifier.
4	NALG	Not supported	L'algorithme n'est pas supporté.
5	NHASH	Not supported	La fonction de hachage n'est pas supportée.
999	NDOC	Not documented	

Chapitre 9

Conclusions et perspectives

Après avoir longuement analysé les différentes solutions à l'intégration de services de sécurité dans le standard EDIFACT, nous avons décrit la série de recommandations des Nations Unies pour la sécurité au niveau du Message UN/EDIFACT.

Cependant, le domaine d'application des recommandations peut être élargi dans deux directions : vers une introduction de la sécurité dans plusieurs niveaux de la structure EDIFACT et vers une gamme plus large de service.

La série de recommandations actuelle se limite principalement à la sécurité au niveau du Message, bien que quelques aspects de la sécurité au niveau de l'Interchange sont abordés dans la recommandation pour la sécurité séparée du Message. Des extensions vers la sécurité intégrée au Message pour le niveau Interchange et pour le niveau segment ainsi que les Messages de service correspondants seront abordés dans les phases suivantes.

La série de recommandations actuelle solutionne l'ensemble des services de sécurité exposés dans le chapitre 3. Le service de confidentialité et sa proposition de Message CIPHER font également l'objet d'une recommandation des Nations Unies. Dans le rapport TEDIS II « security in open environments » qui prend le relais de TEDIS I « digital signatures in EDIFACT », on définit de nombreux nouveaux services de sécurité qui sont basés sur des demandes commerciales précises, nous citerons entre autres :

- *Claim of creation* : le droit d'auteur est un service de sécurité important dans le traitement électronique des documents. Le problème majeur pour renforcer les droits d'auteurs est qu'il est difficile de décider, de deux versions différentes, laquelle est l'original. Ce problème épineux pour des documents électroniques existe également dans le monde du papier.
- *Claim of ownership* : le but est de prouver qu'un document électronique, à n'importe quel moment, est la propriété temporaire d'un utilisateur particulier. Ce service est clairement requis pour des documents commerciaux négociables, les deux exemples les plus intéressants sont le « Bills of Lading », un document qui indique le propriétaire des marchandises d'un cargo, et le paiement électronique.
- *Anonymous registration in data bases* : le but est de maintenir la possibilité de faire des statistiques sur une base de données tout en protégeant la vie privée des personnes enregistrées.
- *Interactive authentification* : pour l'EDI interactif (I-EDI), en plus des services de sécurité classique requis pour l'EDI, les parties impliquées expriment le besoin particulier de s'identifier mutuellement via l'échange d'une série de messages. La réservation de places d'avions est un exemple d'utilisation de l'I-EDI.

Les perspectives pour EDIFACT seront d'intégrer un certain nombre d'applications fondamentales de l'EDI, tel l'I-EDI, qui ne sont pas encore couvertes par EDIFACT mais également des applications beaucoup plus compliquées, tel l'équivalent électronique des documents négociables qui présente la particularité d'être un challenge considérable pour

la cryptographie. L'intégration de telles applications dans EDIFACT ne se conçoit que par la prise en compte des services de sécurité associés.

Ces recommandations des Nations Unies poursuivent des objectifs d'intégration formelle de la sécurité dans EDIFACT. La nature théorique de ce mémoire provient du caractère formel de l'analyse et de ses résultats, l'étendue de cette analyse ayant laissé peu de place aux aspects pratiques. Par exemple, la mise en place d'une autorité de certification pourrait faire l'objet d'un mémoire à elle seule. D'un autre côté, en parallèle avec les développements des recommandations aux Nations Unies, des firmes privées offrent des gammes de produits de sécurité qui couvrent la quasi totalité des aspects pratiques. Pour preuve, la firme Philips présente la ligne de produits ETHOS (Electronic Trade Handling - Office Security) qui permet la signature électronique des messages EDI dans le respect de la norme EDIFACT.

Bibliographie

[AUTACK, 94]
United Nations Economic Commission for Europe :
AUTACK : Secure Authentification and Acknowledgement Message -
Development of United Nations Standard Messages, UNSMs
TRADE/WP.4/R.1026/Add3, Genève, 23 février 1994

[BLO, 91]
S. BLOCH
EDI : Echange de Données Informatisé - Tome 1
Introduction à l'échange de Données Structurées en Syntaxe EDIFACT
Eyrolles, Paris, 1991

[CEC, 92]
Commission of the European Communities & KPMG:
Secure EDI - a Management Overview
Luxembourg : Office for Publications of the European Communities, 1992

[DEC, 94]
B. DECLOUX
Analyse et implémentation de services de sécurité dans une messagerie X400/84.
(mémoire de fin d'étude)
Intstitut d'informatique, FUNDP Namur, 1994

[DID, 90]
United Nations Economic Commission for Europe
United Nations Trade Data Interchange Directory (UNTDID)
Issue 90.1, Genève, 1990

[DAV, 89]
D.W. DAVIES & W.L. PRICE
Security for Computer Networks
An introduction to Data Security in Teleprocessing
and Electronic Funds Transfer (Second Edition)
John Wiley & Sons, 1989

[GEV, 93]
M. D'UDEKEM-GEVERS
Standards EDI de représentation des données
Cahiers de la CITA, FUNDP Namur, novembre 1993

[HEN, 88]
J. HENSHALL & S. SHAW
OSI Explained : End-to-End Computer Communication Standard
Ellis Horwood Limited, 1988, Chichester (UK)

[IMP, 93]
M. HENDRY
Implementing EDI
Artech House, Boston-London, 1993

[ISO-9735]
ISO :
EDIFACT : Application Level Syntax Rules
Genève, 1988

[KAO, 92]
W. C. KAO
Analyse et développement d'un prototype X.435 utilisant X.400 version 1984
(mémoire de fin d'étude)
Institut d'informatique, FUNDP Namur, 1992

[MAR, 93]
A.J. MARCELLA & Jr. and S. CHAN
EDI Security, Control and Audit
Artech House, Boston-London, 1993

[MIG, 94]
United Nations Economic Commission for Europe :
MIG Handbook - UN/EDIFACT Message AUTACK
TRADE/WP.4/R.1026/Add4, Genève, 22 février 1994

[MLS, 94]
United Nations Economic Commission for Europe :
Recommandations for UN/EDIFACT Message Level Security
TRADE/WP.4/R.1026/Add1, Genève, 22 février 1994

[SECUR]
J. RAMAEKERS & J. HUBIN
Sécurité et fiabilité des systèmes informatiques
Syllabus, FUNDP Namur, 1995

[SIG, 94]
United Nations Economic Commission for Europe :
EDIFACT Security Implementation Guidelines
TRADE/WP.4/R.1026/Add2, Genève, 22 février 1994

[SMT, 94]
United Nations Economic Commission for Europe :
Security for UN/EDIFACT Message Transfer
TRADE/WP.4/R.1026, Genève, 7 février 1994

[SOE, 93]
M. DE SOETE
The key to Open EDI : Digital Signature
EEMA Winter Conference, Brussels, janvier 1993

[TED1, 90]
CRYPTOMATHIC A/S
A proposal concerning the use of Digital Signatures in EDIFACT
TEDIS, Brussels, Novembre 90

[TED2, 92]
CRYPTOMATHIC A/S & MBLE
Security in Open Environments
TEDIS II, Brussels, Octobre 92

[X.509]
CCITT :
The directory - Authentication Framework
Recommandations X.509
CCITT, Melbourne, 1988

Annexe I : Acronymes

ANSI	American National Standards Institute
AT&T	American Telephone and Telegraph
BT	British Telecom
CA	Certification Authority
CAD/CAM	Computer Aided Design / Computer Aided Manufacturing
CBC	Cipher Block Chaining
CCITT	Comité Consultatif International pour la Télégraphie et la Téléphonie
CEC	Commission of the European Communities
DEA	Data Encryption Algorithm
DES	Data Encryption Standard
EDCD	UN/EDIFACT Composite Data Element Directory
EDCL	UN/EDIFACT Code List
EDED	UN/EDIFACT Data Element Directory
EDI	Electronic Data Interchange
EDIFACT	voir UN/EDIFACT
EDMD	UN/EDIFACT Standard Message Directory
EDSD	UN/EDIFACT Segment Directory
EFT	Electronic Funds Transfer
ISO	International Standardisation Organisation
I-EDI	Interactive EDI
MAC	Message Authentification Code
MD	Message Development
MDC	Manipulation Detection Code
MIG	Message Implementation Guidelines
ODETTE	Organisation for Data Exchange by TeleTransmission in Europe
OSI	Open Systems Interconnection
PEDI	The commitee developing EDI specifications for the X.400 standards family. Connu également sous le nom de X.435
PIN	Personal Identification Number
PTT	Post Telephone and Telegraph Administration
RSA	Rivest, Shamir and Adelman
RSVA	Réseau de Services à Valeur Ajoutée
RVA	Réseau à Valeur Ajoutée
SIG	Security Implementation Guidelines
SWIFT	Society for Worldwild Interbank Financial Telecommunications
TDI	Trade Data Interchange
TEDIS	Trade Electronic Data Interchange Systems
TRADACOMS	TRAding Data COMmunicationS
TSS	Time-Stamping Service
TTP	Trusted Third Party
UNCID	Uniform rules of Conduct for the Interchange of Data by teletransmission

UN/EDIFACT	United Nations / rules for Electronic Data Interchange For Administration, Commerce and Transport
UNSM	United Nations Standard Message
VAN	Value Added Network
WP.4	Working Party 4

Annexe II :
Security Implementation Guidelines

UNITED
NATIONS

E

Economic and Social
Council

Distr.
RESTRICTED

TRADE/WP.4/R.1026/Add.2
22 February 1994

ENGLISH ONLY

ECONOMIC COMMISSION FOR EUROPE

COMMITTEE ON THE DEVELOPMENT OF TRADE

Working Party on Facilitation of
International Trade Procedures
(Item 7 of the provisional agenda of
the Meetings of Experts on Data Elements
and Automatic Data Interchange (GE.1)
Forty-ninth session, 15-16 March 1994)

EDIFACT SECURITY IMPLEMENTATION GUIDELINES

* * *

Submitted by the Rapporteur for the West European EDIFACT Board *

* The present document is reproduced in the form in which it was received by the secretariat.

GE.94-30557

3. DATA SEGMENT SPECIFICATION

SEGMENT GROUP 1 (Conditional, 9)

USH	Security Header	M	1
USA	Security Algorithm	C	1
	Segment Group 2	C	2
USC	Certificate	M	1

This segment group identifies the security mechanism applied to the message in which the group is included. It includes the identification of the parties involved in the security process (security elements originator and security elements recipient), the reference of the secured message which includes the date of creation of the security elements and the identification of the security algorithm used (including the technical parameters needed by the algorithm). If the algorithm used requires certificates they may also be conveyed in segment group 2 as part of the present segment group 1.

The algorithm identified in the USA segment is the algorithm directly applied to the message content, either a symmetric algorithm (for message origin authentication, or integrity) or a hash function (for non-repudiation of origin).

In the case of non-repudiation of origin by means of a digital signature, the asymmetric algorithm used to produce the signature will be identified within the certificate segment if it is present, or will be implicitly known by the receiving party if the certificate is not conveyed in the message. In this latter case the asymmetric algorithm may be decided in the Interchange Agreement.

3.1 USH - SECURITY HEADER (Mandatory, 1)

3.1.1 Segment Format

Number	Description	M/C	Format	Special notes
0552	SECURITY STRUCTURE VERSION NUMBER	M	an..3	
0501	SECURITY FUNCTION, CODED	M	an..3	
0534	SECURITY RESULT LINK	M	n2	
0541	SCOPE OF SECURITY APPLICATION, CODED	C	an..3	
0503	RESPONSE TYPE, CODED	C	an..3	
0505	FILTER FUNCTION, CODED	C	an..3	
0507	CHARACTER SET ENCODING, CODED	C	an..3	
0509	ROLE OF SECURITY PROVIDER, CODED	C	an..3	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list qualifier	C	an..3	
0515	Code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list qualifier	C	an..3	
0515	Code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0516	SECURITY REFERENCE NUMBER	C	an..35	
S501	SECURITY DATE AND TIME	C		
0517	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	

3.1.2 Segment Description

The SECURITY HEADER segment specifies the security service applied to the message in which the segment is included.

If this segment is used, at least SECURITY STRUCTURE VERSION NUMBER (0552), SECURITY FUNCTION (0501) and SECURITY RESULT LINK (0534) must be present.

The SECURITY IDENTIFICATION DETAILS composite segments in group 1 either :

- must be used if symmetric methods are used (see 3.2.2), or
- may be used if asymmetric methods are used, and if there is a need to distinguish between the security originator certificate and the security recipient certificate

3.1.3 Segment Rules

There may be several different USH segments within the same message, if different security functions are applied to the message (e. g. integrity and non-repudiation of origin) or if the same security function is simultaneously applied by several entities.

The SECURITY RESULT LINK (0534) will be used to establish a link between one USH segment and the related USR segment.

3.1.3.1 SECURITY STRUCTURE VERSION NUMBER (0552)

It specifies the version number of the format of the security headers and trailers identified by year and status of the UN/EDIFACT service segment directory.

3.1.3.2 SECURITY FUNCTION, CODED (0501)

It specifies the security function applied to the message. One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	NRO	Non-repudiation of origin	The message includes a digital signature protecting the receiver of the message from the sender's denial of having sent the message.
2	AUT	Message origin authentication	The actual sender of the message cannot claim to be some other (authorized) entity.
3	INT	Integrity	The message content is protected against the modification of data.

The security functions are described in the document "Recommendations for UN/EDIFACT message level security from the UN/EDIFACT Security Joint Working Group".

3.1.3.3 SECURITY RESULT LINK (0534)

Contains a number which links a particular USH segment with its corresponding UST segment. The value used is arbitrarily assigned but, within one message, the same value must not be used more than once.

3.1.3.4 SCOPE OF SECURITY APPLICATION, CODED (0541)

It specifies the scope of application of the security service defined in the present header, thus it defines the data that have to be taken into account by the related cryptographic process.

This scope may be either :

- the current Security Header segment group (from the first character, namely a "U", to the separator ending this Security Header segment group, both included), and the UNSM body (from the first character following the separator ending the last Security Header segment group to the separator preceding the first character of the first Security Trailer segment group, both included). In this case any other Security Header or Security Trailer segment group will not be encompassed within this scope. The code used in this case is "SHM",

or

- from the first character (namely a "U", included) of the current Security Header segment group to the first character of the related Security Trailer segment group (namely a "U", included). The relation between the Security Header and Security Trailer segment groups is provided by the data elements security result link of the USH and of the UST segments. In this case, the scope of application of the security mechanism of the current header will encompass the UNSM body and all the Security

Headers and Security Trailers embedded within this Security Header and its related Security Trailer. The code to use in this case is "SHT".

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	SHM	Security Header and Message body	see explanation above
2	SHT	From Security Header to Security Trailer	see explanation above

3.1.3.5 RESPONSE TYPE, CODED (0503)

It specifies whether a secure acknowledgment from the message recipient is required or not. If it is required, the message sender will expect an AUTACK message to be sent back by the current message recipient to the current message sender, containing this acknowledgement.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	NA	No acknowledgement required	No AUTACK acknowledgment message expected
2	AC	Acknowledgement required	AUTACK acknowledgment message expected

3.1.3.6 FILTER FUNCTION, CODED (0505)

Identification of the filtering function used for validation results and keys .

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	NUL	No filter	Self explanatory
2	HEX	Hexadecimal filter	Hexadecimal filter
3	ASC	ISO 646 filter	ASCII filter as described in DIS 10126-1 (see note)
4	BAU	ISO 646 Baudot filter	Baudot filter as described in DIS 10126-1
5	EDA	UN/EDIFACT level A filter	UN/EDIFACT level A filter function as described in Annex A of the present document
999	ZZZ	Mutually agreed	Self explanatory

note :
This filtering function is mentioned here for completeness, but does not comply with requirements of UN/EDIFACT level A or B syntax.

3.1.3.7 CHARACTER SET ENCODING, CODED (0507)

Identifies the character set in which the message was coded when security mechanisms were applied.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	AS7	ASCII 7 bit	Self explanatory
2	AS8	ASCII 8 bit	Self explanatory
3	EBC	EBCDIC IBM 360	IBM 360 machine EBCDIC code
4	IPC	ASCII IBM PC	IBM PC ASCII code
5	VAX	ASCII DEC VAX	DEC VAX ASCII 8 bit code
999	ZZZ	Mutually agreed	Self explanatory

3.1.3.8 ROLE OF SECURITY PROVIDER, CODED (0509)

Identifies the function of the security provider as to the secured item.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	ISS	Issuer	The security provider is the rightful issuer of the signed document
2	NOT	Notary	The security provider acts as a notary in relation to the signed document
3	CON	Contracting party	The security provider endorses the content of the signed document
4	WIT	Witness	The security provider is a witness, but is not responsible for the content of the signed document
999	ZZZ	Undefined	The role of the security provider is not defined

note :

when this data element is not used, the value "ISS" is assumed.

3.1.3.9 SECURITY IDENTIFICATION DETAILS (S500)

Identification of parties involved in the security process. Two occurrences of this composite data element are possible : one for the security originator, one for the security recipient.

If asymmetric algorithms are used, party identification is performed by the use of certificates. Hence, this composite should be used either :

- if symmetric algorithms are used, or
- if asymmetric algorithms are used and when two certificates are present, in order to distinguish between the originator and the recipient certificates

If these composite data elements are used at least the Security party qualifier (identifying security originator or security recipient) must be present.

SECURITY IDENTIFICATION DETAILS composite data elements should be used only if the parties involved in security are not un-ambiguously identified by certificates (use of symmetric methods or asymmetric methods, to clarify the use of more than one certificate).

The identification of the SECURITY IDENTIFICATION DETAILS composite data elements related to the security originator and of the one related to the security recipient is achieved by the Security party qualifier.

If asymmetric methods are used, this SECURITY IDENTIFICATION DETAILS composite data element may be used to identify that a certificate is used by an entity acting as security originator or security recipient. In this case, the identification of the party of the SECURITY IDENTIFICATION DETAILS composite data element (any of the data elements 0511, 0513, 0515, 0586) in the USH segment group will be the same as the identification of the party, qualified as "certificate owner" in the USC segment group, and the security party qualifier will identify the function (originator or recipient) of the party involved.

Security party qualifier (0577)

Specification of the function of the security party identified. One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	MS	Message sender	Identifies the party which generates the security parameters of the message (i.e. security originator).
2	MR	Message receiver	Identifies the party which verifies the security parameters of the message (i.e. security recipient).

Key name (0538)

Identification of a key.

The USA segment allows all the information related to the cryptographic mechanism applied to the secured message to be conveyed.This includes the identification of the key, in the case of symmetric algorithms.

If there is no need to convey a USA segment in the secured message (because the cryptographic mechanisms have been agreed previously between the partners), the data element key name (0538) of the USH segment may be used to establish the key relationship between the sending and receiving parties.

Nevertheless, it is stongly reommended to use either the data element key name (0538) in the USH segment, or the data element algorithm parameter value (0532) with the appropriate qualifier in the USA (data element algorithm parameter qualifier, with the code value "KYN"), but not both of them, within the same message Security Header.

Party Id identification (0511)

Code identifying a party involved in the security process, according to a defined registry of security parties.

Code list qualifier (0513)

Code identifying the type of identification used to register the security parties.

Code list responsible agency (0515)

Code identifying the agency in charge of registration of the security parties.

Party name (0586)

Identification of the security party. Three occurences may be used to allow a complete identification.

3.1.3.10 SECURITY REFERENCE NUMBER (0516)

Identification of the message to which security is applied. This identification is security related and may differ from the identification of the message that may appear elsewhere. The reference number may be used to provide message sequence integrity.

3.1.3.11 SECURITY DATE AND TIME (S501)

Security timestamp of the message to which security is applied. This timestamp is security related and may differ from any dates and times that may appear somewhere else in the message. It may be used to provide message sequence integrity.

Date and time qualifier (0517)

Specification of the type of date and time. The following code must be used :

Code	Mnemo.	Meaning	Description
1	STS	Security Timestamp	Security timestamp of the secured messsage

Date (0502)

Specification of the date. Its format must be YYYYMMDD (century included).

Time (0504)

Specification of the time. Its format must be HHMMSS (HH being in 24 hour clock format).

UTC offset (0506)

Offset from UTC standard time. Its format may be either :

XHHMM (X being P for plus or M for minus, followed by the offset, in hours and minutes), or

XY (X being : C = Central, E = Eastern, M = Mountain, P = Pacific, and Y being : D = daylight time, S = Standard time, T = Time).

3.2 USA - SECURITY ALGORITHM

3.2.1 Segment Format

Number	Description	M/C	Format	Special notes
S502	SECURITY ALGORITHM	M		
0523	Use of algorithm, coded	M	an..3	
0525	Cryptographic mode of operation, coded	C	an..3	
0533	Mode of operation code list identifier	C	an..3	
0527	Algorithm, coded	C	an..3	
0529	Algorithm code list identifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier	C	an..3	

3.2.2 Segment Description

This segment is used to identify an algorithm, the technical usage made of it, and to contain the technical parameters required.

The use of algorithm data element specifies the usage made of the algorithm, the algorithm data element identifies the algorithm itself, and the cryptographic mode of operation specifies the mode of operation used.

The mode of operation code list identifier and the algorithm code list identifier identify in which code list the codes of the preceding data elements (mode of operation or algorithm) are defined.

The algorithm parameter composite data element provides space for one parameter. It may be repeated up to 5 times. The number of repetitions actually used will depend on the algorithm used. The order of the parameters is arbitrary but, in each case, the actual value is followed by a coded algorithm parameter Qualifier. Most algorithms in use today will not require parameter values to be the full allowable length.

Where the USA segment is used within a USH segment, the algorithm may be either symmetric, or a hash function.

Asymmetric algorithms shall not be referred to directly in USH segments but may appear only within Segment Group 2, triggered by a USC segment.

3.2.3 Segment Rules

3.2.3.1 SECURITY ALGORITHM (S502)

Use of algorithm, coded (0523)

Specifies the usage made of the algorithm identified by the algorithm data element (0527).

The use of algorithm is required to enable the different SECURITY ALGORITHM composite data elements, used in the different USA segments of Segment Group 2, to be distinguished from each other. In a USH segment the usage of the algorithm is determined by the Security Function qualifier of the USH segment. The use of algorithm must be used with one of the following codes :

Code	Mnemo.	Meaning	Description
1	OHA	Owner hashing	Specifies that the algorithm is used by the message sender to compute the hash function on the message. (as in the case of Non-repudiation of Origin identified in the security function qualifier of USH).
2	OSY	Owner symmetric	Specifies that the algorithm is used by the message sender either for integrity or message origin authentication (specified by Security function qualifier in USH).

Cryptographic mode of operation, coded (0525)

Identifies the mode of operation used, for the algorithm specified by the algorithm (0527) data element.
In the USH segment one of the following codes must be used :

Code	Meaning	Description
0	NUL	Mode of operation meaningless for the current algorithm.
1	ECB	DES modes of operation, Electronic Code Book; FIPS Pub 81 (1981); ANSI X3.106; IS 8372 (64 bits); ISO 10116 (n-bits).
2	CBC	DES modes of operation, Cipher Block Chaining; FIPS Pub 81 (1981); ANSI X3.106; IS 8372 (64 bits); ISO 10116 (n-bits).
3	CFB1	DES modes of operation, Cipher feedback; FIPS Pub 81 (1981); ANSI X3.106; IS 8372 (64 bits); ISO 10116 (n-bits).
4	CFB8	DES modes of operation, Cipher feedback; FIPS Pub 81 (1981); ANSI X3.106; IS 8372 (64 bits); ISO 10116 (n-bits).
5	OFB	DES modes of operation, FIPS Pub 81 (1981); IS 8372 (64 bits); ISO 10116 (n-bits).
6	MAC	Message Authentication Code ISO 8731-1, using DES CBC mode.
7	DIM1	Data integrity mechanism using a cryptographic check function; ISO DIS 9797, first method
8	DIM2	Data integrity mechanism using a cryptographic check function; ISO DIS 9797, second method
9	MDC2	Modification Detection Code - IBM System Journal, vol 30, no 2, 1991.
10	HDS1	Hash functions - Part 2 : Hash functions using a n-bit block cipher algorithm providing a single length hash code. ISO CD 10118-2.
11	HDS2	Hash functions - Part 2 : Hash functions using a n-bit block cipher algorithm providing a double length hash code. ISO CD 10118-2.
12	SQM	Square-mod n hash function for RSA. Annex D, CCITT X 509. ISO 9594-8.
13	NVB7.1	Dutch Standard hash function for banking.
14	NVBAK	Dutch Banking Standard, NVB Authenticity Mark, published by the NVB, May'1992.
999	ZZZ	Mutually agreed.

Mode of operation code list identifier (0533)

Specification of the code lists used to identify the cryptographic mode of operation. When the codes defined above, by the UN/EDIFACT SJWG, as published in the present document, are used the value "1" must be used.

Algorithm, coded (0527)

Identifies the algorithm. In the USH segment one of the following codes must be used :

Code	Meaning	Description
1	DES	Data Encryption Standard. FIPS Pub 46 (January 1977).
2	MAA	Message Authentication Algorithm. Banking-Approved Algorithms for message Authentication. ISO 8731-2.
3	FEAL	FEAL Fast Data Encipherment Algorithm.
4	IDEA	International Data Encryption Algorithm : Lai X., Massey J. "A Proposal for a New Block Encryption Standard", Proceedings of Eurocrypt'90, LNCS vol 473, Springer-Verlag, Berlin 1991, and Lai X., Massey J. "Markov Ciphers and Differential Cryptanalysis", Proceedings of Eurocrypt'91, LNCS vol 547, Springer-Verlag, Berlin 1991.
5	MD4	The MD4 Message digest algorithm. Rivest R. RSA Data Security Inc. (1990).
6	MD5	The MD5 Message digest algorithm. Rivest R. Dusse S. RSA Data Security Inc. (1991).
7	RIPEMD	Extension of the MD4 - Ripe Report CS - R9324, April 93.
8	SHA	Secure Hashing Algorithm.
9	AR/DFP	Hash function of the German banking industry, submitted to ISO/IEC JTC 1/SC 27/WG 2, Doc N179.
999	ZZZ	Mutually agreed.

The presence of a particular algorithm in the list does not imply any endorsement of that algorithm.

The above algorithms are those currently in common use (November 1993). It is expected that new algorithms will be added as they become available, and are generally accepted.

Algorithm code list identifier (0529)

Specification of the code lists used to identify the algorithm. When the codes defined above by the UN/EDIFACT SJWG, as published in the present document, are used the value "1" must be used.

3.2.3.2 ALGORITHM PARAMETER (S503)

Algorithm parameter value (0532)

This component contains the value of a parameter required by the algorithm referenced in the algorithm data element. The precise type, usage and format of the value is specified in the immediately following algorithm parameter qualifier (0531). If necessary, this value is filtered by the filter function identified in the FILTER FUNCTION, CODED data element (0505) of the USH segment (key names do not need to be filtered).

Algorithm parameter qualifier (0531)

Identifies the type of the algorithm parameter value that immediately precedes it.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	IVC	Initalisation Value, cleartext	Identifies the algorithm parameter value as an unencrypted initialisation value.
2	IVE	Initialisation Value, encrypted under a symmetric key	Identifies the algorithm parameter value as an initialisation value which is encrypted under the symmetric data key.
3	IVP	Initialisation Value, encrypted under a public key	Identifies the algorithm parameter value as an initialisation value encrypted under the public key of the receiving entity.
4	IVZ	Initialisation Value, format mutually agreed	Identifies the algorithm parameter value as an initialisation value in a format agreed between the two parties.
5	KYE	Symmetric key, encrypted under a symmetric key	Identifies the algorithm parameter value as a symmetric key which is encrypted with a previously agreed algorithm under a previously exchanged symmetric key.
6	KYP	Symmetric key, encrypted under a public key	Identifies the algorithm parameter value as a symmetric key encrypted under the public key of the receiving entity.
7	KYS	Symmetric key, signed and encrypted	Identifies the algorithm parameter value as a symmetric key signed under the sender's secret key, then encrypted under the receiver's public key.
8	KYD	Symmetric key encrypted under an asymmetric key common to the sender and the receiver	Identifies the algorithm parameter value as a symmetric key encrypted under an asymmetric key common to the sender and the receiver (use of Diffie and Hellmann scheme, for instance).
9	KYN	Symmetric key name	Identifies the algorithm parameter value as the name of a symmetric key. This may be used in the case where a key relationship has already been established between the sender and receiver.
10	KKN	Key encrypting key name	Identifies the parameter value as the name of a key encrypting key.
11	KYZ	Symmetric key, format mutually agreed	Identifies the algorithm parameter value as a symmetric key in a format agreed between the two parties.
999	ZZZ	Parameter value is mutually agreed	Identifies the algorithm parameter value as having a usage and format that is mutually agreed.

SEGMENT GROUP 2 (Conditional, 2)

USC	Certificate	M	1
USA	Security Algorithm	C	3
USR	Security Result	C	1

When asymmetric algorithms are used, this segment group contains the data necessary to validate the security methods applied to a message.

In its most common form, the certificate will include the public key and the credentials of the certificate owner signed by the Certificate Issuer.

A certificate may either be conveyed completely (the USC segment, 3 USA segments and the USR segment), or may be conveyed only as a Certificate Identifier (the USC segment identifying the Certificate), to refer to a public key that is already known by the entities involved or retrieved from a data base.

Two occurrences of this Segment group are allowed, one being the message sender certificate (that the message receiver will use to verify the message sender's signature), the other being the message receiver certificate (presumably only referred to by Certificate Reference) in the case where the receiver public key is used by the sender for confidentiality of symmetric keys.

If two certificates (sender's and receiver's) are simultaneously present within one security header, the Security Identification Details data element (S500) together with the Certificate Reference data element (0536) allow them to be differentiated.

3.3 USC - CERTIFICATE (Mandatory, 1)

3.3.1 Segment Format

Number	Description	M/C	Format	Special notes
0536	CERTIFICATE REFERENCE	C	an..35	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list qualifier	C	an..3	
0515	Code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
S500	SECURITY IDENTIFICATION DETAILS	C		
0577	Security party qualifier	M	an..3	
0538	Key name	C	an..35	
0511	Party Id identification	C	an..17	
0513	Code list qualifier	C	an..3	
0515	Code list responsible agency	C	an..3	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0586	Party name	C	an..35	
0544	FORMAT CERTIFICATE VERSION	C	an..3	
0505	FILTER FUNCTION, CODED	C	an..3	
0507	CHARACTER SET ENCODING, CODED	C	an..3	
0543	CHARACTER SET REPERTOIRE, CODED	C	an..3	
0546	USER AUTHORISATION LEVELS	C	an..35	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S505	SEPARATOR FOR SIGNATURE	C		
0548	Separator for signature	C	an..4	
0551	Separator for signature qualifier	C	an..3	
S501	SECURITY DATE AND TIME	C		
0513	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	
S501	SECURITY DATE AND TIME	C		
0513	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	
S501	SECURITY DATE AND TIME	C		
0513	Date and time qualifier, coded	M	an..3	
0502	Date	C	n8	
0504	Time	C	n6	
0506	UTC offset	C	an..5	

3.3.2 Segment Description

USC segments will be needed when public key cryptography is used, even if certificates are not used. Either the full certificate is present (including the USR segment), or the only data

elements of the certificate are the Certificate reference (0536) and the SECURITY IDENTIFICATION DETAILS (S500) composite data element identifying the isuer Certification Authority or the SECURITY IDENTIFICATION DETAILS (S500) composite data element identifying the Certificate Owner, including its public key name. The presence of a full certificate may be avoided if the certificate has already been exchanged by the two parties.

3.3.3 Segment Rules

3.3.3.1 CERTIFICATE REFERENCE (0536)

Uniquely identifies one certificate for a Certification Authority. This field may be used to refer to a certificate when the whole certificate is not conveyed. The unique certificate reference may be obtained by the Certificate reference (0536) and the SECURITY IDENTIFICATION DETAILS (S500) composite data element identifying the Certification Authority.

3.3.3.2 SECURITY IDENTIFICATION DETAILS (S500)

Identification of parties involved in the certification process.

Two occurrences of this composite data element are possible : one for the Certificate Owner (identifying the entity which signs the message), one for the Certificate Issuer (Certification Authority or CA).

When this composite data element is used at least the Security Party Qualifier must be present.

The identification of Certificate Owner and Certification Authority is achieved by the data element Security Party Qualifier.

Security party qualifier (0577)

Specification of the function of the security party identified. One of the following codes must be used :

Code	Mnemo.	Meaning	Description
3	OW	Certificate Owner	Identifies the party which owns the Certificate.
4	AX	Authenticating party	Party which certifies that the document (i. e. the certificate) is authentic.

Key name (0538)

Identification of a public key : either the public key of the owner of this certificate, or the public key related to the secret key used by the Certificate Issuer (CA) to sign this certificate. In the latter case, this field allows a Certification Authority to use several keys, either for separate applications, or consecutive generations of CA keys.

Party Id identification (0511)

Code identifying a party involved in the security process, according to a defined registry of security parties. In the USC segment this party may be either :

- the party which owns the Certificate (Certificate Owner), or
- the party which certifies that the document (i. e. the certificate) is authentic (Authenticating party : CA)

Code list qualifier (0513)

Code identifying the type of identification used to register the security parties.

Code list responsible agency (0515)

Code identifying the agency in charge of registration of the security parties.

Party name (0586)

Identification of the security party. Three occurences may be used to allow a complete identification.

3.3.3.3 FORMAT CERTIFICATE VERSION (0544)

Version number of the version of the certificate, identified by year and status of the UN/EDIFACT service segment directory.

3.3.3.4 FILTER FUNCTION, CODED (0505)

Identification of the filtering function used for validation results and keys, within the certificate.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	NUL	No filter	Self explanatory
2	HEX	Hexadecimal filter	Hexadecimal filter
3	ASC	ISO 646 filter	ASCII filter as described in DIS 10126-1 (see note)
4	BAU	ISO 646 Baudot filter	Baudot filter as described in DIS 10126-1
5	EDA	UN/EDIFACT level A filter	UN/EDIFACT level A filter function as described in Annex A of the present document
999	ZZZ	Mutually agreed	Self explanatory

note :

This filtering function is mentioned here for completeness, but does not comply with the requirements of UN/EDIFACT level A or B syntax.

3.3.3.5 CHARACTER SET ENCODING, CODED (0507)

Identifies the character set in which the certificate was coded when security mechanisms were applied to it.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	AS7	ASCII 7 bit	Self explanatory
2	AS8	ASCII 8 bit	Self explanatory
3	EBC	EBCDIC IBM 360	IBM 360 machine EBCDIC code
4	IPC	ASCII IBM PC	IBM PC ASCII code
5	VAX	ASCII DEC VAX	DEC VAX ASCII 8 bit code
999	ZZZ	Mutually agreed	Self explanatory

3.3.3.6 CHARACTER SET REPERTOIRE, CODED (0543)

Identifies the syntax level used to create the certificate, when security mechanisms were applied to it.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	UNOA	UN/EDIFACT syntax level A	Self explanatory
2	UNOB	UN/EDIFACT syntax level B	Self explanatory
3	UNOC	UN/EDIFACT syntax level C	Self explanatory
4	UNOD	UN/EDIFACT syntax level D	Self explanatory
5	UNOE	UN/EDIFACT syntax level E	Self explanatory
6	UNOF	UN/EDIFACT syntax level F	Self explanatory

3.3.3.7 USER AUTHORISATION LEVELS (0546)

Specification of the privileges, authorisation level, etc. associated with the owner of the certificate.

3.3.3.8 SEPARATOR FOR SIGNATURE (S505)

Identifies the characters used as syntactical separators between all components or within composite components, of the USC segment when the signature was computed. The syntactical separators used in the message may be different to these characters. The data element Separator for signature qualifier (0551) identifies each separator. If the composite data elements SEPARATOR FOR SIGNATURE (S505) are not present, the syntactical characters used are those currently used in the message.

Separator for signature (0548)

Separator used when the signature was computed. In order to avoid translator problems, this separator is represented by its value in the character set identified by the CHARACTER SET ENCODING data element (0507), hexa-filtered on, at least, two characters. For example the separator "'" is coded "27" (two characters), ":" is coded "3A" (two characters), the release character "?" is coded "3F" (two characters) and the separator "+" is coded "2B", if ASCII 8bit code page is used.

Separator for signature qualifier (0551)

Identifies each separator (either separator between data element or separator within a composite). One of the following codes must be used :

Code	Mnemo.	Meaning	Description
1	SEG	Segment separator	Specifies that this is the separator between segments.
2	DAT	Data separator	Specifies that this is the separator between data elements.
3	COM	Composite separator	Specifies that this is the separator within a composite data element.
4	REL	Release character	Specifies that this is the release character.

3.3.3.9 SECURITY DATE AND TIME (S501)

Identification of the dates and times involved in the certification process.

Three occurrences of this composite data element are possible : one for the certificate generation date and time, one for the certificate start of validity period , one for the certificate end of validity period.

The distinction between the certificate generation date and time, the certificate start of validity period and of the certificate end of validity period is achieved by the Date and time qualifier.

Date and time qualifier (0515)

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
2	CGT	Certificate generation time	Identifies the date and time of generation of the certificate by the Certification Authority.
3	CSV	Certificate start of validity period	Identifies the date and time from which the certificate must be considered valid.
4	CEV	Certificate end of validity period	Identifies the date and time until which the certificate must be considered valid.

Date (0502)

Specification of the date. Its format must be YYYYMMDD (century included).

Time (0504)

Specification of the time. Its format must be HHMMSS (HH being in 24 hour clock format).

UTC offset (0506)

Offset from UTC standard time. Its format may be either :

XHHMM (X being P for plus or M for minus, followed by the offset, in hours and minutes),
or
XY (X being : C = Central, E = Eastern, M = Mountain, P = Pacific, and Y being : D = daylight time, S = Standard time, T = Time).

3.4 USA - SECURITY ALGORITHM (Mandatory, 3)

3.4.1 Segment Format

Number	Description	M/C	Format	Special notes
S502	SECURITY ALGORITHM	M		
0523	Use of algorithm, coded	M	an..3	
0525	Cryptographic mode of operation, coded	C	an..3	
0533	Mode of operation code list identifier	C	an..3	
0527	Algorithm, coded	C	an..3	
0531	Algorithm code list identifier	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier, coded	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier, coded	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier, coded	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier, coded	C	an..3	
S503	ALGORITHM PARAMETER	C		
0532	Algorithm parameter value	C	an..512	
0531	Algorithm parameter qualifier, coded	C	an..3	

3.4.2 Segment Description

This segment is used to identify an algorithm, the technical usage made of it, and to hold the technical parameters required.

The use of algorithm data element specifies the usage made of the algorithm, the algorithm data element identifies the algorithm itself and the cryptographic mode of operation data element specifies the mode of operation used.

The mode of operation code list identifier and the algorithm code list identifier identify the code list in which the codes of the preceding data elements (mode of operation or algorithm) are defined.

The algorithm parameters composite data element provides space for one parameter. It may be repeated up to 5 times. The number actually used will depend on the algorithm used. The order of the parameters is arbitrary but, in each case, the actual value is followed by a coded algorithm parameter qualifier. Most algorithms in use today will not require parameter values to be the full allowable length.

In Segment Group 2, triggered by the USC segment, three USA segments are present. These are :

1. the algorithm used by the Certificate Issuer to compute the hash value of the Certificate (hashing function)

2. the algorithm used by the Certificate Issuer to generate the Certificate (i. e. to sign the result of the hash function computed on the certificate content) (asymmetric algorithm)

3.a - either the algorithm used by the sender to sign the message (i. e. to sign the result of the hash function described in the USH segment, computed on the message content) (asymmetric algorithm),

3.b - or the receiver's asymmetric algorithm used by the sender to encrypt the key required by a symmetric algorithm applied to the message content and refered to by the Segment Group 1 triggered by the USH segment (asymmetric algorithm),

These three occurrences of the USA segment are distinguished by the Use of algorithm data element (0523).

3.4.3 Segment Rules

3.4.3.1 SECURITY ALGORITHM (S502)

Use of algorithm (0523)

Specifies the usage made of the algorithm identified by Algorithm data element (0527). In a USA segment within a USC one of the following codes must be used :

Code	Mnemo.	Meaning	Description
3	ISG	Issuer signing	Specifies that the algorithm is used by the Certificate Issuer (CA) to sign the hash result computed on the certificate
4	IHA	Issuer hashing	Specifies that the algorithm is used by the Certificate Issuer (CA) to compute the hash result on the certificate
5	OCF	Owner enciphering	Specifies that the algorithm is used by the message sender to encrypt a symmetric key
6	OSG	Owner signing	Specifies that the algorithm is used by the message sender to sign either the hash result computed on the message or the symmetric keys
7	OCS	Owner enciphering or signing	Specifies that the algorithm is used by the message sender to encrypt a symmetric key or sign the hash result computed on the message

Cryptographic mode of operation, coded (0525)

Identifies the mode of operation used, for the algorithm specified by the algorithm (0527) data element.
In the USC segment one of the following codes must be used :

Code	Meaning	Description
0	NUL	Mode of operation meaningless for the current algorithm.
9	MDC2	Modification Detection Code - IBM System Journal, vol 30, no 2, 1991.
10	HDS1	Hash functions - Part 2 : Hash functions using a n-bit block cipher algorithm providing a single length hash code. ISO CD10118-2.
11	HDS2	Hash functions - Part 2 : Hash functions using a n-bit block cipher algorithm providing a double length hash code. ISO CD10118-2.
12	SQM	Square-mod n hash function for RSA. Annex D, CCITT X 509. ISO 9594-8.
15	MCCP	Banking key management by means of asymmetric algorithms, Algorithms using the RSA cryptosystem. Signature construction by means of a separate signature. ISO CD 11166-2.
16	DSMR	Digital Signature Scheme Giving message recovery. ISO 9796.
999	ZZZ	Mutually agreed.

Mode of operation code list identifier (0533)

Specification of the code lists used to identify the cryptographic mode of operation. When the codes defined above by the UN/EDIFACT SJWG, as published in the present document, are used the value "1" must be used.

Algorithm, coded (0527)

Specifies the algorithm. In the USC segment one of the following codes must be used :

Code	Meaning	Description
1	DES	Data Encryption Standard. FIPS Pub 46 (January 1977).
5	MD4	The MD4 Message digest algorithm. Rivest R. RSA Data Security Inc. (1990).
6	MD5	The MD5 Message digest algorithm. Rivest R. Dusse.S RSA Data Security Inc. (1991).
7	RIPEMD	Extension of the MD4 - Ripe Report CS - R9324, April 93.
8	SHA	Secure Hashing Algorithm.
9	AR/DFP	Hash function of the German banking industry, submitted to ISO/IEC JTC 1/SC 27/WG 2, Doc N179.
10	RSA	Rivest, Shamir, Adleman: A Method for obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM, Vol.21(2), pp 120-126 (1978).
11	DSA	Digital Signature Algorithm/Digital Signature Standard NIST Pub 1993 Draft.
12	RAB	Rabin, "Digitalized signatures and public-key functions as intractable as factorization", MIT Laboratory for Computer Science Technical Report LCS/TR-212, Cambridge, Mass, 1979.
999	ZZZ	Mutually agreed.

Algorithm code list identifier (0529)

Specification of the code lists used to identify the algorithm. When the codes defined above by the UN/EDIFACT SJWG, as published in the present document, are used the value "1" must be used.

3.4.3:2 ALGORITHM PARAMETER (S503)

Algorithm parameter value (0532)

This component contains the value of a parameter required by the algorithm referenced in the algorithm data element. The precise type, usage and format of the value is specified in the immediately following algorithm parameter qualifier. If necessary, this value is filtered by the filter function identified in the FILTER FUNCTION data element (0505) of the USC segment (key names do not need to be filtered).

Algorithm parameter qualifier (0531)

Identifies the type of the algorithm parameter value that immediately precedes it.

One of the following codes must be used :

Code	Mnemo.	Meaning	Description
12	MOD	Modulus	Identifies the algorithm parameter value as the modulus of a public key which is to be used according to the function defined by the use of algorithm.
13	EXP	Exponent	Identifies the algorithm parameter value as the exponent of a public key which is to be used according to the function defined by the use of algorithm.
14	MLN	Modulus Length	Identifies the algorithm parameter value as the length of the modulus (in bits) of the public key used in the algorithm. The length is independent of whatever filtering function may be in use.
15	PR1	Generic parameter 1	Identifies the algorithm parameter value as the first generic parameter (see note)
16	PR2	Generic parameter 2	Identifies the algorithm parameter value as the second generic parameter (see note)
17	PR3	Generic parameter 3	Identifies the algorithm parameter value as the third generic parameter (see note)
18	PR4	Generic parameter 4	Identifies the algorithm parameter value as the fourth generic parameter (see note)
19	PR5	Generic parameter 5	Identifies the algorithm parameter value as the fifth generic parameter (see note)
20	PR6	Generic parameter 6	Identifies the algorithm parameter value as the sixth generic parameter (see note)
21	PR7	Generic parameter 7	Identifies the algorithm parameter value as the seventh generic parameter (see note)
22	PR8	Generic parameter 8	Identifies the algorithm parameter value as the eighth generic parameter (see note)
23	PR9	Generic parameter 9	Identifies the algorithm parameter value as the ninth generic parameter (see note)
24	PRA	Generic parameter 10	Identifies the algorithm parameter value as the tenth generic parameter (see note)
999	ZZZ	Parameter value is mutually agreed	Identifies the algorithm parameter value as having a usage and format that is mutually agreed.

note :
These generic parameters are provided to allow the use of any algorithm requiring identification of parameters different from the parameters defined above.

When the DSA algorithm (NIST, Pub 1993) is used, PR1 contains parameter "P", PR2 contains parameter "Q", PR3 contains parameter "G", PR4 contains parameter "Y".

3.5 USR - SECURITY RESULT (Mandatory, 1)

3.5.1 Segment Format

Number	Description	M/C	Format	Special notes
S508	VALIDATION RESULT	M		
0560	Validation value	M	an..256	
0560	Validation value	C	an..256	

3.5.2 Segment Description

The USR segment included in the USC segment contains the signature computed by the Certification Authority by signing the hash result computed on the data of the credentials.

In the case of signature algorithms requiring two parameters to express the result, the two data elements validation result are used in an order described in the documents about these algorithms.

3.5.3 Segment Rules

3.5.3.1 VALIDATION RESULT (S508)

Validation value (0560)

This component contains the digital signature computed by the Certification Authority on the data of the credentials. This signature is computed using first the hash function defined by the qualifier "issuer hashing" ("4") in the use of algorithm data element, then the asymmetric algorithm defined by the qualifier "issuer signing" ("3") in the use of algorithm data element.

The digital signature is computed according to the rules and with the parameters specified in the USC segment (CHARACTER SET ENCODING, SEPARATOR FOR SIGNATURE, CHARACTER SET REPERTOIRE). The signature computation starts with the first character of the USC segment (namely a "U") and ends with last character of the last USA segment (including the separator following this USA segment).

This data element is filtered, if necessary, by the filter function identified in the FILTER FUNCTION data element (0505) of the USC segment.

The length of this data element is determined by the length of the key (one of the Algorithm parameter data elements, qualified by the Algorithm parameter qualifier "modulus length" ("14"), of the Issuer signature algorithm) and the filter function applied to the result of the signature process.

In the case of RSA signature, only one validation value data element is used.

In the case of DSA signature two validation value data elements are required. The first one corresponds to the parameter known as "r", the second one to the parameter known as "s".

SEGMENT GROUP n (Conditional, 9)

	Segment Group 3	C	9
UST	Security Trailer	M	1
USR	Security Result	C	1

This segment group contains the link with the related USH segment and the security result corresponding to the security functions specified in this USH segment. For every security trailer (Segment Group n, triggered by UST) there is one corresponding security header (Segment Group 1, triggered by USH).

3.6 UST - SECURITY TRAILER (Mandatory, 1)

3.6.1 Segment Format

Number	Description	M/C	Format	Special notes
0534	SECURITY RESULT LINK	M	n2	

3.6.2 Segment Description

This segment is used to separate the UNSM body from the security trailer.

The UST segment contains a number which links the UST segment with its corresponding USH segment.

3.6.3 Segment Rules

3.6.3.1 SECURITY RESULT LINK (0534)

Contains a number which links a particular USH segment with its corresponding UST segment. The value used is arbitrarily assigned but, within one message, the same value must not be used more than once.

3.7 USR - SECURITY RESULT (Conditional, 1)

3.7.1 Segment Format

Number	Description	M/C	Format	Special notes
S508	VALIDATION RESULT	M		
0560	Validation value	M	an..256	
0560	Validation value	C	an..256	

3.7.2 Segment Description

Contains the security result corresponding to the security functions specified in the linked USH segment

In the case of signature algorithms requiring two parameters to express the result, the two data elements validation result are used in an order described in the documents about these algorithms.

3.7.3 Segment Rules

3.7.3.1 VALIDATION RESULT (S508)

Validation value (0560)

Contains the security result corresponding to the security functions specified in the linked USH segment.

This data element is filtered, if necessary, by the filter function identified in the FILTER FUNCTION data element (0505) of the USH segment.

The length of this data element is determined by the length of the key (one of the Algorithm parameter data elements, qualified by the Algorithm parameter qualifier "modulus length" ("14"), of the Owner signature algorithm) and the filter function applied to the result of the signature process.

In the case of RSA signature, only one validation value data element is used.

In the case of DSA signature two validation value data elements are required. The first one corresponds to the parameter known as "r", the second one to the parameter known as "s".

Table des matières